HUMAN MOVEMENT UNDERSTANDING From Computational Geometry to Artificial Intelligence

ADVANCES IN PSYCHOLOGY 33 Editors G. E. STELMACH P.A.VROON cW C 111 n NORTH-HOLLAND AMSTERDAM NEW YORK OXFORD TOKYO

HUMAN MOVEMENT UNDERSTANDING From Computational Geometry to Artificial Intelligence Edited by P. MORASSO and V.TAGLIASCO Department of Communication, Computer and Systems Science University of Genoa Genoa, Italy . ._ AKAOl?" \Zy VAN £7, 1986 NORTH-HOLLAND AMSTERDAM NEW YORK • OXFORD TOKYO

 ISBN: 0444700323 Publishers: ELSEVIER SCIENCE PUBLISHERS B.V. P. O. Box 1991 1000 BZ Amsterdam The Netherlands Sole distributors for the U.S.A. and Canada: ELSEVIER SCIENCE PUBLISHING COMPANY, INC. 52 Vanderbilt Avenue New York, N.Y. 10017 U.S.A. Library of Congress Cataloging-in-Publlcation Data Human movement understanding. (Advances in psychology j 33) Includes index. 1. Human mechanics—Research—Technique. 2. Kinesiology—Research—Technique. 3. Human mechanics —Mathematical models. 1*. Artificial intelligence. I. Morasso, P. (Pietro) II. Tagliasco, V. (Vincenzo) III. Series. QP301.H83 1986 612'.76 86-9021 ISBN O-UUU-70032-3 PRINTED IN THE NETHERLANDS

V La somma e principals parte dell'arte e' I'invenzione de' com- ponimenti di qualunque cosa; e la seconda parte e' de' movlmenti che abbiano attenenza alle loro operazioni, le quali sieno fatte con prontitodine, secondo i gradi de' loro operatori, cosi' in pi- grizia come in sollecitudine ... I movimenti dell'uomo sopra un medesimo accidente sono infini- tamente vari in se' medesimi ... Se le figure non fanno atti pronti i quali colle membra esprimano il concetto della mente loro, esse figure sono due volte mode, perche' morte sono principalmente che' la pittura in se' non e' viva, ma esprimitrice di cose vive senza vita, e se non le si aggi- unge la vivacita' dell'atto, essa rimane morta la seconda volta ... Leonardo da Vinci Trattato della Pittura - Codice Urbinate Vaticano

VI ACKNOWLEDGEMENTS We should like to thank Emilio Bizzi - from whom we got the essential initial message - for providing constant intellectual support and inspiration to our work, Neville Hogan for participating in the experiments reported in the chapter on compliance, and Giovanni Degll Antoni for his invaluable suggestions on the idea of using Petri nets for musical purposes. We are also greatly indebted with a great number of students who contributed to many parts of the experiments and software development and, mostly important, were a basic element of the creative and investigative environment which is the source of research. The following Is just an incomplete list: A. Batistoni, M. Canepa, C. Crocco, Q. Qrasso, A. Kanitz, F. Lavagetto, Q.P. Marcenaro, Q. Nlcolosi, F. Pasio, M. Parodi, M. Pellissetto, M. Pilotto, R. Rocca, C. Raggio, M. Solari, M. Sanguineti, M. Tistarelli, M. Tonioni, Q. Vercelli, A. Viacava. The research reported in this book has been supported by the Ministry of Education (National Programs on Artificial Intelligence, Robotics, Bioengineering), by the Canter of Bioen- gineering and Anthropomorphic Robotics of the National Research Council, and by the ESPRIT Programme (AIP-P418 Image and Movement Understanding).

VII PREFACE This "work in progress" stems from fifteen years of "struggle" with the subtle complexities of human movement. Its development, although not always rewarding, has been fascinating. Perhaps this is a common experience for all the people who are trying to understand human movement. Indeed, several investigators have attempted to look at movement in a "global" way with a higher horizon than the standard narrow scope imposed by the constraints of experimental activity. In general, we may distinguish between two approaches: horizontal and vertical. In the horizontal case, one may try to collect and integrate the views and the knowledge that can be found in different "movement disciplines" from motor neurophysiology to sport science. In the vertical case, the effort is focused on making associations among classes of problems and methodologies which deal with relevant aspects of movement, although they may not be "movement disciplines" per se. This book is undoubtedly tilted in the second direction and it aims to express the distributed expertise about "action", "environment", "shape", and "time" that was possible to find in our young department. The readers will not enjoy a systematic account of human movement research, but a coordinated collection of methodological tools that we think useful for those who are fascinated by movement and skills and wish to start experimenting with motion organization concepts in a sufficiently general way. Pietro Morasso and Vincenzo Tagliasco Qenova, September 30, 1985

VIII LIST OF CONTRIBUTORS G. Adorni L. Baratto R. Boccardo A. Camurri S. Gaglio A. Grattarola G. Haus G. Marino L. Massone F. Mussa Ivaldi C. Ruggiero G. Sandini V. Tagliasco E. Troiano P. Vacchina R. Zaccaria L Baratto is with the Center of Motor Education, USL 16, Via G. Maggio, I - 1614B Genoa, Italy. G. Haus is with the Institute of Cybernetics, University of Milan, Via Viotti 5, I - 20133 Milan, Italy. F. Mussa Ivaldi, when writing his contribution, was with the Psychology Dept., M.I.T., Cambridge, Massachusetts. He is now at DIST. All the other contributors are with DIST (Department of Communication, Computer and Systems Science), University of Genoa, Via Opera Pia 11,1- 16145 Genoa, Italy.

CONTENTS

PROLEGOMENA Human movements are complex patterns which cannot be understood by the laws of physics even if they are constrained by physics in every detail. The falling rock or the thrown rock follow trajectories which can be explained by using the fundamental laws of physics and by applying them to the peculiarities of the particular experiment. The complexity of the description is a function of the level of detail/precision that we, as observers of the phenomenon, wish to obtain: we may be content with the explanation provided by elementary mechanics, which treats the rock as a point mass or we may further and further complicate (and improve) it by taking into account gas dynamics, thermodynamics, phenomena at the molecular and atomic level, and so on. However, if the level of precision/detail that we desire is the one that characterizes common experience, then the thrown rock can be safely modeled as a point mass and its motion can be perfectly understood in the framework of elementary Newtonian mechanics. On the contrary, the human body is a much more complex object even at the same empiric level of detail - a level at which we can model it as a puppet. The degree of complexity of such a model can be measured by the number of degrees of freedom, which Is close to a hundred. Also the "trivial" application of elementary Newtonian mechanics to such a mechanical object results in a surprisingly complex formulation which, at any rate, helps us little to understand human movements. In fact, there Is a significant difference between "understanding" the motion of a rock (or of a more complicated object, like a wobbling puppet) and "understanding" the motion of the human body (or of a puppet animated by a skilled puppet showman). This is the difference between the point of view of physics and the point of view of cybernetics. The former one starts from the set of forces acting on a body, whioh are considered known (gravity, pressure, friction, etc.) and tries to account for the resulting movement: How is the rock going to fly after I have thrown it? Therefore, the emphasis is on energy and on dynamics. The point of view of cybernetics is just the opposite: it starts from the definition of the goal (a desired behaviour) and then it tries to force the system to follow it, possibly using a large range of control techniques, either of the feedback or of the feedforward type. The emphasis, in thia case, is on information, not on energy: sensory information, motor information, information processing, abstract representation of information structures. The cybernetic approach leads logically, as it did historically, to face two main classes of topics: (i) Automatic Control topics, such as stability, observability, and controllability, which are particularly concerned with the enforcement of a prescribed behavior, and (ii) ArtificiaJ Intelligence

 topics which are centered around the problem of the {arbitral}/) construction and representation of knowledge. (In a sense, this distinction mirrors the eternal dualism between law and freedom.) Although Knowledge Representation is a "hot" topic, attracting a considerable amount of attention in a number of different areas, very few attempts have been made to help understand human movements, which means to build up a theory of Motor Knowledge. The aim of this book is in this direction and it is based on two working assumptions: 1. it Is meaningful to separate the physical, control, and knowledge aspects of human movements, and 2. the latter is the appropriate level at which "we" can understand movement and master its complexity 1. It must be stressed that the notion of motor knowledge Is far from purely theoretical: It Is computational in the sense of Man- [1982] and it must also be stressed that "we" of the previous sentence refers to investigators from a number of different areas: neuroscience, sport science, dance, motor rehabilitation, study of motor development, computer animation, and robotics. We can summarize the argument with two provocative examples: Moving humans and puppets animated by skilled puppet showmen even though characterized by very dlt- lereni physical and control aspects, basically require the same type of motor knowledge. The problem ol movemant notation/representation is basically the sema in writing the script ot a dance ballet, lhe program ot a computer animated humanold, the control program ot a robotic manipulator arm, or in describing with a symbolic motor script the movements ol a child or ot a motor disabled person. Regarding motor problems, the difference between the physic approach and the cybernetic one can be traced back to the concept of causality, as it is conceived In common sense as well as in classic western philosophy. In the latter case, for example, the Aristotlean distinction between an efficient cause and a final cause mirrors the distinction between the internal and external forces that determine a movement and the motor representation which shapes the movement 2. A corollary of this point of view is that it stresses the geometric and kinematic aspects of human movements, rather neglecting the phenomena at the joint and muscular levels. In this sense, what motor knowledge is about is just shape, that is, structure In space and time. This suggests an association of motor knowledge with geometric modeling and vision which are usually considered quite uncoupled topics. The association between form and movement is obvious, for example in handwriting, because the result of the movement, the script, is a visible geometric entity, but it underlies every kind of motor behavior such as grasping objects, making grasped objects interact, or navigating among objects. ' We wish to stress that we are not simply disregarding physics and control theory (how sometimes A.I. scientists are tempted to do). Physic and control aspects are relevent to our purpose to the extent that they are represented and used at the level ot motor knowledge. The second Chapter ot part II. tor example, deals with the hend compliance and its dependence on lhe viscous elastic properties ot muscles, with the purpose ol representing It at the cognitive level. 2 The distinction is somewhat blurred il we consider variational problems, tor example the motion ot a body in a conservative field ot loroas. In this case, the behavior ol the body can be explained by its "goal* ol reaching a configuration of minimum potential energy. However, 'shaping* the lietd according to a task is a problem of knowledge representation.

Prolegomena 3 Planning motor actions requires to "match" in a general sense, the shape of the movement to the shape of the environment, either the "real" shape of objects or the "virtual" shape of the free space inside which the human body moves. For example, In handwriting, or drawing, or painting, two tasks must be concurrently carried on: one is the task of driving the hand/pencil/brush along a trajectory with a desired shape (the trajectory formation task). The other is the task of applying an appropriate pressure, which is an attempted but constrained motion against the writing surface, with an appropriate compliance (the compliance modulation task). Both tasks have a natural geometric characterization. Certainly, task planning should take Into account also form independent knowledge such as the influence of lightness, fragility, texture, and color, which is a typical claim of Al scientists. However, as we deem important to consider motor knowledge independently of physical and control aspects, we think that form- independent knowledge corresponds to a quite distinct domain of Al that we are not going to consider in this book. We previously brought up the "computational" point of view which characterizes our attempt to outline a theory of motor knowledge. The meaning of this adjective is twofold: in one (modeling) sense it refers to the concept of the motor system as an information processing system characterized by different levels of representation and by their mutual transformations; in another (methodological) sense, "computational" points out the fact that general methods of motor knowledge, far from being purely theoretic, can be applied to very different application fields (dance, computer animation, rehabilitation etc.) avoiding ad hoc methods and "tricks" which are standard but unsatisfactory. The book is organized into four parts: Part I: action, Part II: environment, Part III: shape, Part iv: time. The first part is focused on the organization and representation of motor patterns. Chapter 1 deals with trajectory formation, i.e. the process of forming trajectories by means of simple "primitive" movements with a standard spatio-temporal structure. Chapter 2 discusses the possible extension of these concepts to understand more complex motor patterns such as walking, which entail a highly structured spatio-temporal organization. Chapter 3 examines the different techniques of "movement and dance notation" stressing the underlying computational complexity, and Chapter 4 discusses the NEM project (the NEw Men) which aims at providing a global programming environment for representing and animating motor patterns and actions. The second part relates to the interface between the performing human and his environment. In particular, Chapter 1 discusses at length the concept of compliance (how it may be represented and how it is related to the mechanical properties of muscles) and Chapter 2 deals with problems of navigation and planning. (It includes examples from the SMASH project: Soft Manoevering Among SHapes.) The third part presents concepts and techniques for representing objects. Chapter 1 considers methods for the analysis and the synthesis of "smooth" shapes. (It includes a discussion of

 the SCULPTOR-2 shape editor.) Chapter 2 discusses some aspects of the interaction between shape and action. The fourth part relates to another of the foundations of "motor patterns": the co-ordination of concurrent parallel procasses. General concepts and methods are presented In Chapter 1 and a significant "case study" Is discussed in Chapter 2 for the representation of musical structures by means of Petri nets (the MAP system: Musical Actors by Petri nets). The common emphasis which permeates most of the contributions is the issue of "representation", i.e. the need to express in a formal way different aspects of movement and action together with the need of having an homogeneous framework for transforming one representation into another and for expressing communication among representations. The underlying concept is that movement and action are extremely multi - faceted and understanding them is mainly a problem of formalizing an appropriate network of representations. Perhaps, the shift from a musculo - centered and/or retino - centered approach to a representation - centered one in the investigation of skilled behavior is a Copernican revolution of which we have not yet appreciated the full range.

PART I ACTION This part contains four chapters which explore different aspects of the computational structure of actions. The chapters are the following: Chapter 1: Trajectory Formation Chapter 2: Complex Motor Patterns Chapter 3: Dance and Movement Notation Chapter 4: NEM: a language for the Representation of Motor Knowledge The first chapter examines simple movements that require the motor coordination of two or more degrees of freedom (e.g. eye and head, elbow and shoulder, etc.). It is shown that in many cases the movement structure is dominated by the spatial location of the target (the goal of the movement), not by the biomechanical aspects. The process of trajectory formation is investigated by means of mathematical models which emphasize the concept of concurrency in motor control. The chapter is terminated with two sections which summarize the fundamental kinematic and dynamic equations of kinematic chains. The second chapter, developing the modeling concepts of the previous chapter, attempts a formalization of complex motor patterns such as locomotion. This formalization consists of a set of concurrent interacting automata which express the different functional aspects of the walking skill. The goal is to capture the organizing structure which remains invariant in different walking circumstances: locomotion in different directions and with different trajectories with or without obstacles etc. The second part of the chapter looks at pathological motor behaviour, particularly as regards walking, in the case of central lesions of the nervous system. Classical syndroms of walking impairment are reviewed and the underlying problems of motor organization/disorganization are examined from the particular point of view of motor rehabilitation. The third chapter deals with examples of movement formalization which were motivated by the contingent need to code a motor practice, in dance and gymnastics, for example. Several notation systems are examined and particular attention is paid at two systems: the Eshkol- Wachman system and the codification of movement in Tai Chi Ch'uan. Although these formalizations are not scientific in the strict posltivistic term, they are quite valuable from the point of view of (motor) intelligence and (motor) knowledge representation. The final part of the chapter outlines a computational approach to movement notation and representation, which exploits the technical and epistemological opportunities provided by the contemporary computer era. The last chapter deals with a computer project that is being developed at DIST for the representation and simulation of complex movement patterns (the NEM project).

This Page Intentionally Left Blank

PARTI Chapter 1 TRAJECTORY FORMATION Pietro Morasso CONTENTS

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasco (Editors) © Elsevier Science Publishers B.V. (North-Holland), 1986 TRAJECTORY FORMATION Pietro Morasso For a computational approach to the information processing aspect of intelligent behavior, motor activity is classified as a multiplicity of action paradigms such as muscle contractions, joint rotations, trajectories, and tasks with no rigid hierarchy among them. Different elements of trajectory formation including co-ordination, timing, and geometric transformations are considered here with regard to motor activities of increasing complexity: eye head movements for target acquisition, arm targeted movements, and complex trajectory formation patterns. The uniform spatio-temporal structure which is found in the large range of motor paradigms motivates the hypothesis of "a kinematic planning of fundamental motor patterns" on the basis of a "principle of smaller complexity of the representation*. A trajectory formation model is then presented, that explores in computational terms the kinematic planning hypothesis, and specific modeling aspects of handwriting are considered, that aim at "normalization" and "coding" of cursive script. Finally, the kinematic and dynamic transformations that underly trajectory formation are reviewed together with

the corresponding mathematical formalism. 1. THE COMPUTATIONAL APPROACH From the beginning of computer science a few decades ago, similarities and dissimilarities between brains and computers have been a topic of controversial discussion. The comparison between brains and oomputers will become more earnest as computers are applied to a greater and greater extent to activities which were formerly regarded as essentially human. At the same time, the expectation of future "intelligent" computers and robots is encouraging a new approach to the study of intelligence and purposive behavior, which is leading to a new science: We may call It in many equivalent ways such as "human Information processing" or "anthropomorphic robotics". In any case, the emphasis is on the computational structures which may lead to a certain level of performance. If we compare brains and computers, at first sight we may find more differences than similarities. For example, regarding anatomy, the "hardware" - neurons and microcircults - have

 very little in common in physical characteristics, speed, type and number of inputs and outputs, etc., unless we consider them simply as basic information processing blocks. However, the history of computer science itself teaches us that drastic technological revolutions do not affect essential computational concepts. A common unbroken line joins the mechanical Pascal or Bab- bage machines with the electromechanical calculators of the last century; with ENIAC, the first electronic computer that used the technology of thermolonic valves; with the different generations of modern computers that exploit the tremendous potential of semiconductors. The story is obviously still in progress and we may expect new and different technologies without necessarily having to change the whole concept of "computer" and/or of "computation". Also the basic arithmetic and information codes are probably different. For example there is no reason to think that the brain operates with binary numbers In the same way in which modern computers do. But also in this case we may argue that the specific choice of a code is not essential. We might have computers with different codes and different types of Information storage and still performing the same type of computational functions. Indeed, different types of codes are used in the computer industry. For example, the BCD code is sometimes used instead of the pure binary code for numbers. Different codes are used for representing characters (ASCII, EBCDIC, Hollerith, etc.) and a number of different codes and protocols exist for Interconnecting computers with peripheral equipment and/or other computers.

The difference between brains and computers is even more striking as regards the "architecture". From what is known of the brain, we may assume that it processes information in a parallel way along millions of ohannels which, though rather slow, operate concurrently and asynchronously. On the contrary, existing electronic computers process Information sequentially at a very fast rate. However, this kind of architecture (the so called von Neuman architecture1) is far from being the ultimate design paradigm and indeed, It is being lively challenged as the applications of the computer have expanded from pure data processing. Image processing, A.I., and robotics are examples of fields where new computational architectures are being theorized and implemented. However, beyond such a long list of differences between brains and computers, Is there a level at which we can find similarities and permanent features? If we think that the answer Is yes, this is clearly the level of the task that brains and computers perform, which is an Information processing task. The Independence of Information processing tasks from the specific hardware which implements them has been forcefully discussed by David Marr [Marr 1962] in relation with human and computer vision. However, the same concepts which have been formulated by Marr are equally applicable to movement. In general terms, the goal of vision Is to "make explicit what is in the world and where" and the goal of movement Is to "navigate In the world and manipulate objects". Both tasks can be carried out using different types of hardware and different architectures. The computational approach consists of stressing the information processing aspects of 1 The negative judgment within the A.I. community which is associated with the "von Neuman" architecture does not do justice to von Neuman who, as one ol the founders ol Cybernetics (Wiener 1851), was well aware of the parallel nature of the brain architecture. In aplie of that, "electronic brains' were and are still developed as sequential machines.

Trajectory Formation 11 intelligent behavior and abstracting them from specific hardware and architectures. The computational approach, in our opinion, is not only a very useful epistemological tool for understanding intelligent behavior, but it is also a distinctive strategic characteristic of Intelligence itself. For example, the basic motor patterns of "picking up a banana" should be represented and programmed independently of the initial conditions of a specific "banana pickup" such as size, shape, position, and orientation of the banana; and Initial posture, selection of supporting limbs and of the grasping limb, and activation of specific groups of muscles. Similarly, the visual prooessing of optical data, such as the retinic image of a banana, cannot be reduced to a pure pattern matching, and it must be independent of the particular shape, color, position, and orientation of a banana. 2. MUSCLES, JOINT ROTATIONS, TRAJECTORIES, TASKS As pointed out in the previous section, "movement" (as well as "vision") can mean many different things. In particular, it is possible to make the following examples of movement paradigms: (a) Muscles contractions without motion, e.g. isometric movements of one part of the body against another one or against an object, with the purpose of exerting a force. (b) Rotations of a body part around a joint, e.g. rotations of the eyeball in the eyesocket, of the head around the neck, or of the hand around a doorknob. (c) Trajectories of body parts with regard to the body (e.g. in eating "the hand brings the food to the mouth") or with regard to some object in the world (e.g. in handwriting the handheld pen "traces a curve on a piece of paper"). (d) Fundamental motor tasks such as walking, where the main sub-task, "navigate the body in the world along a path", is performed concurrently with other instrumental tasks (such as "keep from falling" (which in turns require other and more specific sub - sub - tasks (such as "co-ordinate the legs and the arms" (which most likely will depend strongly on the specific "walking performance" (nature and shape of the terrain, type of shoes etc.))))). (e) Complex motor tasks, such as assembly line work or dancing, where all the previous motor patterns may come into action, in different combinations and sequences. Therefore, even if all movements end up eventually with patterns of muscular activities, they can be best understood in terms of rotations, trajectories or complex combinations of them, according to the type and the context of the motor performance. Furthermore, muscular effer-ences aione do not identify uniquely a movement, because muscles are neither "pure force actuators" nor "pure length actuators". They are indeed "compliant actuators* [Mussa Ivaldl et al. 1984], Thus, movement emerges from the interaction between "muscular efference" and the "environment" [Bernstein 1987].

 The ecological nature of perception has been strongly emphasized by Gibson [Gibson 1979] and the same ecological concept has been extended to action by Reed [Reed 1984], Turvey and Kugler [Turvey and Kugler 1984], and others. The key aspect, from our point of view, is that perception and action are not uniquely determined by afference and efference but also by the flow of energy which links an actor and its environment. From this follow the theories of "direct perception" and "direct action" which oppose the conventional "retino-centric" or "musculo- centric" approach. According to these theories, perception and action are conceived as "direct" relations between the actor and his environment. Even if the direct approach has been contrasted to the computational approach [Ullman 1980] as antagonistic, in our opinion the direct approach only implies a refusal of retino - centric or musculo - centric computations. On the contrary, the central role of the relation between the actor and the environment, which is emphasized by the direct approach, points out the predominance, for example, of kinematics with respect to dynamics, and of solid modeling with respect to image processing. The previous considerations explain why, in our view, muscle activity patterns are not appropriate to characterize movement. Similarly, joint rotations are in most cases "internal events" far removed from the direct actor - environment interface. Head/eye movements however, are a notable exception. They are worth considering on their own because they offer a good example of the stereotyped spatio-temporal patterns which might be used by the brain as basic building blocks of action synthesis. 3. EYE-HEAD MOVEMENTS FOR TARGET ACQUISITION The movements of the eyes have the function of rotating the optic axis with respect to the head, in such a way that a visual target is either acquired or maintained in the central area of the retina. Therefore, in the case of the eyes, "movement" means "rotation" and the "target grasping" function of the eye movements is complemented by the movements of the head. From a mechanical point of view, eyes and head are a kinematic chain, one of the many that can be considered in the human body. The motion of the eye with respect to the environment Is the sum of the eye rotation (with respect to the head) and of the head rotation (with respect to the environment) [Jones and Milsum 1965]. The function of the eye head chain, as of other chains, is to relate in a desired way the last element of the chain (the eye in our case) with some feature of the environment. This implies a geometric mapping between "global" variables and "internal" variables. In general, the mapping is non-linear, but this is not the case for the eye head system. What is also special in the eye head system Is the vestibulo- ocular reflex (VOR), an automatic mechanism which drives the eye rotations in such a way as to compensate for voluntary or involuntary head movements. The VOR operates in feedforward from inertial sensors (the semicircular canals) that, at the same time, stabilize in feedback the position of the head in space. This kind of arrangement tends to cut out the mechanical coupling of the eye head chain allowing the brain to program movements of the eyes - in - space without bothering

Trajectory Formation 13 about the "disturbing" effect of the head motion. In general, dominating the mechanical coupling is the purpose of any controller of a kinematic chain. What is special about the eye head chain with respect to other chains such as the arm, is that the functional uncoupling is obtained by a measurement (the reafferent vestibular signal) Instead of a patterned motor program, as it is more likely to happen in the case of arm movements. There are functional justifications for this special arrangement related to the strategic role played by the vestibular organ in all the orientation matters, particularly those which cannot be pre-programmed in navigation tasks. However, it is interesting to point out that the eye head system can also perform in the more "standard" pre-programmed way, if the special vestibular sensor is not available. This was demonstrated by recording pre-programmed compensatory movements of the eye in labyrinthectomized monkeys, after a suitable period of recovery [Dichgans et al. 1973] (see figure 3.1). Thus, even though it is different from the other kinematic chains of the body for its mechanical structure and for its normal mode of operation, the eye head system seems to have access to the same brain structures which allow to program in a coherent way all the degrees of freedom of a kinematic chain. Let us now consider two interesting aspects of the eye head co-ordination for their relation with the other kinematic chains of the body. One has to do with the shape of "eye head trajectories" and the other with their temporal structures. (We are considering here only the saccadic movements, which are analogous to the reaching/pointing/touching movements of the hand.) Regarding the shape of the eye head trajectories in the exploration of the environment, it has been found to be approximately straight (see figure 3.2). This result must be considered in ABC G 600 msec Figure 3.1. Eye Head Co-ordinated Targeted Movements in Intact and Labyrinthectomized Monkeys. The curves represent eye rotations (E), head rotations (H), gaze rotation (G: E + H). (A) shows the normal patterns. (Note the sacoadic eye movement followed by a compensatory movement which keeps the gaze on target.) (B) shows the patterns observed aner labyrynthectomy. (Note the gaze overshoot and the consequent backward saccade) (C) shows the apparently normel patterns after a suitable recovery. (Note that the appropriate compensstory movements are driven by the vestlbuio ocular reflex in the intact animal and are pre-programmed in the vestlbulectomized animal.) [Dichgans et al. 1973].

14 P. Morasso Figure 3.2 Trajectories of the Eyes in Scanning a Picture of a Face [Palmieri et al. 1971]. relation with the complex non-linear mapping that exists between the lines of action of the eye/head muscles and the degrees of freedom of the eye [Carpenter 1977, Robinson 1982]. In our opinion, this is one of the pieces of experimental evidence (as it is also argued in the section on arm trajectories) that support the concept of "kinematic planning of fundamental motor patterns" on the basis of a "principle of smaller complexity of the representation". With regards to the temporal structure, it is well known that the eyes, unlike most other motor subsystems, are not characterized by an approximate isochrony, i.e. the duration of saccadic movements is not independent of the movement amplitude but it is a monotonic function [Robinson 1964, Morasso et al. 1973]. There are reasons for both types of performance. The saccadic motor system is a servomechanism that operates at maximum speed and constant load, saturating the power output of the extra-ocular muscles. On the contrary, the neck or arm motor systems are normally operated with a large power reserve that allows implementing an "isochronous strategy" (see figure 3.3). The advantage of isochrony is the simplicity of synchronization of sequences of overlapped movements. It allows generation of arbitrarily complex trajectories with stereotyped motion primitives. (Figure 3.4 shows an example of a sequence of overlapped head saccades generated concurrently with a sequence of eye saccades.) Even if eye and head exemplify two different temporal patterns it is worth noting the kind of "temporal matching" that exists between the duration of head movements (about half a second) and the refractory time of eye saccades (about a quarter of a second). In such a way, two saccades (a main saccade and a corrective one) can fit into a single head saccade and a sequence of overlapped head saccades, such as those of figure 3.4, can be nicely synchronized with a concurrent sequence of eye saccades in a coherent exploratory pattern.

Trajectory Formation 15 Figure 3.3. Timing of eye and head saccades of different amplitudes (top: movements to a 20 degrees target, middle: 30 degrees target, bottom: 40 degrees target). Eye and head saccades during targeted movements start simultaneously. The duration of the eye saccades increases systematically with amplitude. The head saccades are substantially isochronous. Time calibration: 100 ms. Vertical calibration: 10,20,30,40 degrees (the left calibration bar refers to the eyes and the right calibration to the head).

16 P. Morasso Eyes Head Head velocity 100 /sec 0 /sec Figure 3.4. Eye-head scanning movements (time calibration: SOO ms) |Blzzi et al 1972], Note modulation of head velocity synchronous with sacoadic eye movements and the temporal matching between the duration of the head movements and the delay of the eye saccades.

Trajectory Formation 17 4. ARM MOVEMENTS FOR TARGET ACQUISITION The acquisition of targets is one of the basic visuo - motor functions. From the visual point of view, it entails co-ordinated eye head movements which bring the optic axis of the eyes onto the target while keeping the eyes in a comfortable position in the head. From the hand point of view, target acquisition means to generate a co-ordinated set of movements which carry the hand from an initial position to the target position. The importance of this simple problem was pointed out by Piaget, who showed how young children's thinking is dominated by the consideration of the point of arrival rather than lengths of time or journeys [Piaget 1946]. The pattern of targeted arm movements depends on the position of the target. If the target is dose enough, then movements of one arm only are functionally necessary, otherwise movements of the trunk and legs are required, sometimes preceded by locomotion movements. On the other hand, in the case of simple movements of the arm, a large number of other motor subsystems are called into action in order to guarantee postural stability [Polit and Massion 1979]2. Therefore, when we speak of "targeting" we mean in general a complex array of concurrent processes. However, if we assume that such processes are independent to a great extent and are co-ordinated on a different computational level, it is still meaningful to focus only on the arm movements induced by targeting tasks in which the distance of the targets is at arm - length. Let us further assume that the fingers and the wrist are restrained from moving as well as the trunk. (In any case, if one has to point at something which is close one will not move his fingers/wrist/trunk significantly.) Thus, the "primitive targeted" movements that we are considering are movements of the shoulder and of the elbow, one of the simplest examples of "multi-joint movements". Holmes [1930] referred to movements which involve several joints as "compound movements" and called the disturbances in joint synchronization seen in patients with cerebellar disorders "decomposition of movement". In the two-joint movements that we are considering, the "decomposition" can be observed if we ask (or we force) a subject to move only one joint at a time, e.g. the elbow first and then the shoulder. The trajectories of the hand that we observe are two circular arcs. If the two joints are rotated together, then an infinite variety of shapes can be obtained, strictly dependent upon the synchronization among the two joints. In targeted movements, only the initial and final positions of the arm are specified by the task. The trajectory which joins them is a choice of the Central Nervous System. In order to investigate such a choice, experiments have been performed [Morasso 1981] in which human subjects were instructed to point one hand to different visual targets which were randomly sequenced in a plane (figure 4.1). 2 The dynamics of the human body determines a croes coupling among all the degrees ot freedom due to gravity, mertial and Corlolis lorces. Therefore, even II only one degree ot freedom is planned to move, all the other degrees ot freedom must ba activated in order to compensate lor the disturbances due to the crces - coupling. However, such a compensation does not necessarily require a detailed prediction of the time course of the disturbing loroes: a rough estimate ot the order ot magnitude ot ihe disturbance can be used to "stiffen up" the muscles to a sufficient extent.

18 p, Morasso Figure 4.1. Arm Targeted Movements in the Plane, (fi.theta) are the intrinsic co-ordinates. (x,y) are the extrinsic coordinates. (T1....T6) are the visual targets used to elicit the targeted movements [Morasso 1981]. The results of the experiments are presented in figures 4.2 and 4.3. Figure 4.2 shows the spatial trajectories of the hand and figure 4.3 shows the time course of the joint angular position, joint angular velocity and hand tangential velocity for four different movements. Joint angular velocities (figure 3, second row) for these different movements exhibit quite different patterns; some joint angular velocities are single peaked and others are double peaked which means inversion of the joint motion. The four movements of the figure exemplify all the four possible combinations (single peak/double peak) for the two joints. In contrast with this, the tangential hand velocity for the different movements has a single peaked curve (an acceleration phase, followed by a deceleration phase) whose shape varies very little from one movement to another. The duration of the different movements was rather constant, independent of the movement distance and of the joint angular pattern. This

kind of "isochrony" of arm targeted movements mirrors the behavior of head targeted movements and is in accordance with previous observations [Bryan 1892, Stetson and McDill 1923]. Furthermore, the trajectories of the hand appear to be approximately straight, a finding corroborated by other researchers [Abend et al. 1982, Hollerbach and Flash 1982]. This finding may be expected on the basis of casual observation of human behavior. It is of interest because the straight hand movements result from the interplay of the rotation of each of the involved joints (elbow and shoulder in our case). Curved or sinuous movements would result if the two degrees - of - freedom were not perfectly co-ordinated. Summing up, the experimental findings on arm targeted movements reveal that invariant features of that pattern are the bell - shaped velocity profile of the hand, and the (rough) straight- ness of the hand trajectory. On the contrary, the joint rotation patterns do not show significant invariances: they are more variable and, therefore, more complex. Other researchers have investigated the opposite possibility, looking for regularities and invariances in the joint rotation patterns, for targeted movements in a vertical plane [Soechting

 "» -V Figure 4.2. Typical trajectories (tor three subjects) of the arm in the plane (the sampling time was 100 samples/sec) (Morasso 1861]. suet i *t (L«0W Hh'r vr ::_ » \J Figure 4.3. Intrinsic and extrinsic patterns during arm targsted movements. Top row: joint rotations. Middle row: joint angular velocities. Bottom row: hand linear velocity. The left column corresponds to a radial movement (from a position dose-central to a position tar-central). The central column to a movement from a position erase-left to a position close-right. The right column corresponds to a movement from* position close-left to far-right (Morasso 1061].

 Morasso and Lacquaniti 1981, Lacquaniti and Soechting 1962]. However, the apparent straight lines in joint space that they found have been demonstrated by Hollerbach and Atkeson [1985] to be artifacts of movement kinematics near the workspace boundary. Let us now consider in greater detail the stereotyped spatio - temporal patterns of the hand in targeted movements. A straight trajectory implies a linear relation between the Cartesian components (x and y) of the hand position vector: (4.1) ax+by+c«0 From this comes the ratio between the two time derivatives (x' and y') that is constant: (4.2) y' / x' - k where the constant k determines the "slope" of the movement, (k equals -a/b.)

This implies, in particular, that x and y (both functions of time) must have the same velocity profile, which must be the same as the velocity profile of the hand v: (4.3) v - (x,2+ y'2)"2 - -(x,2+k2x'2),/2-x'(l +k2)1'2 Therefore, all three functions (x'(t), y'(t), v(t)) can be expressed as multiples of a common temporal function b(t)3, which is bell-shaped and subtends a unitary area. (The proportionality factors represent the distance covered, respectively, in the x direction, in the y direction, and in direction of the movement - see figure 4.4.) As a consequence, the observed targeted movements can be described as a smooth Interpolation of both Cartesian components from the initial position to the final position by means of the same interpolating function: (4.4) X " X|nit * (Xfln " X|nit) 8(t) y - yinit ♦ <y«n - w*> where s(t) is the integral of b(t). (It is a smooth step-like function.) During straight movements of the hand, the ratio between the two joint angular velocities is far from constant, as it is shown in the following (see also fig. 4.1): x - I C, + L C , (4.5) a ' D eT V - 'a Sf ♦ 'b Sef (I and L are the lengths of the humerus and forearm, respectively; e and f are the theta and fi angles; C, is a shorthand notation for cos(fi), C . for cos(theta + fi), etc.) 3 Several mathematical models exist to represent the bell-shaped b(t) function. For example, the cubic B-spline, which uses polynomial segments, and the 'raised cosine", which consists ol one period of a cosine lunction added to a step with the same duration and the same amplitude.

Trajectory Formation 21 ' I -I i U.I ,1 Figure 4.4. Simulated Straight Trajectory. Note that the x and y components of the trajectory have a similar time courses and the corresponding velocities reach the peak at the same time. (4.6) X' " "'a Sf r " 'b *« (8' + n " "V f " 'b Sef 8' / - +la Cf f + lb Cef (•• ♦ f) - +x f + lb Cef e1 from which we get (4.7) Vie- - -(xVy1 lb Cef + lb S^xTy' x +y) and we can see that this ratio changes its value in a complex way except for the very special : ; - (that ►• • • d to those particular movements whose directions pass through the shoulder) where (Cef,SeJ) co-vary with (x,y). In < -■ ■■ , the non - linear mapping which exists > - the "extrinsic" space of the hand (x,y) and the "Intrinsic" - •- : of the joints (fi, theta) has the consequence that a linear trajectory in one •: -- Implies a curved trajectory In the other, and vice versa. The features of arm targeted movements that are discussed above were found also for three-dimensional movements [Morasso 1983]. Figure 4.5 shows a few 3D movements with the >• ::•• >ing velocity and curvature profiles. 4.1. Complexity Let us now discuss about the Issue of complexity which was outlined above. Su • • • ■: that the width of the frequency spectrum of a signal is taken as a measure of its complexity. Then, if

22 P. Morasso •■■•■•■■• f Figure 4.5. Three-Dimensional Arm Targeted Movements. The little spheres (diameter 2 mm) represent samples of the trajectory (sampling rate: 100 samples/8). V: tangential velocity (calibration 200 mm/s). C: curvature (calibration 0.03 1/mm). Duration ot the displayed movement: 8 s. |Morasso 1983]. we consider the joint rotation signals that correspond to the :►•• = • hand = • = = • traj = • = , we can say that the intrinsic re • - - tation of the targeted movements is more complex than the extrinsic = • =; = tation of the same movements because the joint variables oscillate "more quickly" than the - • ►; variables. The brain controls the movements generating a set of efferent signals which are certainly closer to the Intrinsic ; • :. ntatlon than to the extrinsic one. Furthermore, the Intrinsic re' : - tation is the Input to the non-linear - -> =<ic transformation (4.5) which yields the extrinsic - • ■■■■■ tation as its output. The output of a non linear mapping is in general more complex than Its Input because more frequency components tend to be created, unless the input ie "crafted" In such a way that some of the frequency components of the spectrum are mutually - - > out. Although this may happen "by chance", If it is a robust characteristics of motor performance, as it Is, it can only mean that the "motor controller" which generates the complex patterned efference is dominated by a "motor planner" which formulates and shapes a trajectory plan "directly" In terms of the extrinsic ■ • - - - tation. As already anticipated In the previous section, we would like to call this the principle of smaller complexity. The advantage of planning the extrinsic representation Instead of the Intrinsic = • = • tation Is the same, for example, as programming In the Pascal language Instead of In an assembler language: It allows to formulate a larger generality of (motor) problems in a hardware independent way. There is, of course, a computational cost. In the case of Pascal programming, an efficient Pascal compiler is n • •. In the case of the motor planner what Is • • is a neural module

Trajectory Formation 23 which transforms "extrinsic plans" into "intrinsic joint patterns". Such a module would perform what is known as "inverse kinematics" [Benati et al. 1980, 1982). It is still unknown which part of such a computation might be carried out by the muscles in an analogic way (as a result of the programmable compliance of the muscles) and what part of it might come from an explicit "neural computation". The kind of concepts that are discussed above are not completely specific of the motor system. The "direct planning" of hand trajectories, which implies an inverse kinematic computation, is mirrored by the "direct perception" of object shape, which implies an inverse perspective computation. 4.2. Redundance (Generalized Targeted Movements) In the previous discussion of targeted movements we restricted ourselves to two intrinsic degrees of freedom - the non redundant case with elbow and shoulder rotations - versus two extrinsic degrees of freedom (the planar co-ordinates of the hand). However, as already mentioned, targeting may involve a much more complex intrinsic motor pattern if the target is not within arm length. In particular, we would like to report on some yet unpublished data which are concerned with reaching for visual targets on a vertical plane (a large blackboard which was faced by standing subjects). Some of the targets were within arm length and could be reached with neglecteble movements of the body, whereas others were far enough to force the subjects to move their body, shifting their center of gravity sideways and/or vertically, but not so far to make them step aside. Figure 4.6 shows some typical targeted movements, displaying the trajectories of the hand and of the shoulder and their corresponding velocity profiles4. The hand trajectories preserve the same global patterns that characterize the non - redundant case independently of the concurrent body motion revealed by the extent of the shoulder trajectory. In the oase of large movements, the main targeting task Is logically decomposed into three processes: (i) the process which brings the hand on the target, (ii) the process which displaces the shoulder in such a way that the arm does not reach its mechanical limits, and (iii) the process which keeps the balance of the body. The experimental data show that these processes are not activated in a sequential fashion: The movement of the shoulder does not start when the arm is approaching its mechanical limits. On the contrary, body and arm co-operate in a smooth and concurrent way. (The velocity profiles of hand and shoulder in the large movements appear to be started simultaneously.) 4 Six targets were used, three within arm length and three beyond that. The recording technique, which used a TV camera and real lime target tracking, Is the sema discussed in |Morasso 1983).

 ikikL^.A ■> Figure 4.6. Generalized Targeted Movements. The targets are on a blackboard. The right trajectories are drawn by the hand. The left trajectories are drawn by the shoulder while moving the body (sampling frequency: 50 Hz). The extent ol the trajectories is about 1.5 meters. The two plotted time (unctions are the tangential velocity ol the hand (the top one) and the tangential velocity ol the shoulder (the bottom one). Vertical calibration: 1.2 m/s. Horizontal calibration: 1 s. With regard to the body motion which causes the dlapl —; : t of the shoulder, the .:. : i of the action leaves a freedom of choice because the constraint to be met is only a scalar one. (The target must be at arm length.) In fact most of the subjects appear to move their bodies In such a way that the shoulder Is dl • = ► ■ In the direction of the target (the shoulder and hand trajectories »• • on the target). One subject however exhibited a different strategy: The shoulder and hand tr • ■ • • to be rather parallel. The difference •; ■ n the two strategies Is, In general, the dl : -- - between absolute and relative • • ■ - computations. The co-operation between body and arm Is quite Intricate, probably with both feedforward and feedback contributions, but the dominant factor seems to be the planned targeting trajectory of the hand. With regard to the Inverse kinematic function, which must be performed by the motor controller in order to implement the planned trajectory, the data aay that whatever the particular "algorithm" used by the brain, it must be sensitive to the real time variations of the shoulder position which In general are largely unpredictable, as they result from the Interaction with the other concurrent • >-■■■ .

Trajectory Formation 25 5. COMPLEX TRAJECTORY FORMATION PATTERNS Since • - • hand movements reveal a tendency of subjects to produce straight hand paths, it is interesting to study motor paradigms in which the subjects are required to program not only the total extent of the movements but aleo their "shape". This is the case, for example, of 0) targeted movements which are constrained by an obstacle or by a guiding path (we may call them "constrained targeted movements"), (if) "hand gestures" which produce hand trajectories of different shapes, and (iii) handwriting movements. Let us now consider these movement classes in greater detail. 5.1. Constrained T • ■- : • Movements The movement strategy used by subjects who had been instructed to reach a target on a curved path were investigated by Abend et al [1982] and by Morasso and Mussa Ivaldl [1982]. Several experiments were conducted where trajectories of the hand were recorded In the horizontal plane. Figure 5.1 shows a typical -: = of movements made by a subject instructed to move his hand to the target by way of curved paths. The movements • = the same two targets are characterized by an increasing deviation from the straight line. For each movement, the path of the hand, the time course of the hand speed and the path curvature are presented. The first movement is a "pure = • = -•" movement with a near-straight, i.e. zero curvature, path and a bell-shaped velocity profile. In the curved movements, the velocity profile is no longer bell • • and It exhibits a characteristic segmentation. Furthermore, the curved paths appear to be composed of a - of gently curved segments which meet at more highly curved regions. (Consider, for example, the two peaks of curvature of the fourth movement of figure 5.1 which Figure 5.1. Targeted Movemerus Along Curved Trajectories. Sampling frequency: SO Hz. Velocity profile (V: calibration 500 mm/s). Curvature prolite (C: calibration .008 1/mm). Time calibration: 3.5 s |Morasso and Mussa Ivaldl 1982].

P. Morasso > t*ft»M4ft** c u Figure 5.2. Three - Dimensional Curved Movements Performed on Different Planes. The little spheres (diameter 2 mm) represent samples ol the trajectory (sampling rate: 100 samples/s). V: tangential velocity (calibration 200 mm/s). C: curvature (calibration 0.03 1/mm). (x,y,z): components ol the binormal unit vector, i.e. the vector normel to the 'osculating' plane. Duration ol the displayed movement: 8 s. [Morasso 1983]. Figure 5.3. Hand Trajectories Produced While Attempting to Mimic Circular Guide Paths. For each movement the velocity prolile is shown (calibration: 200 mm/s) together with the curvature profile (calibration: 1/150 1/mm). Movement 1 Is a standard targeted movement. Movements along the curved guides are characterized by curvature peaks and temporally associated speed valleys (movements 4 and 6) or inflection points (movements 2 and 3). Sampling frequency: 100 Hz. Time calibration: 500 ms |Abend et al. 1982].

Trajectory Formation 27 r» ■ \ ft Figure 5.4. Hand trajectories to targets whose location can be switched during the movement. Sampling frequency: 100 Hz. Sampling window: 1.280 s. The right part shows the hand trajectories and the left part shows the corresponding velocity profiles (vertical calibration: 1 m/s). Movement C is the re = : >- undisturbed movement. i ! Ti • i LIT* US ft** "•••U t Figure 5.5. Targeted movements at different speeds. Sampling frequency: 100 Hz. Samp l window: 2.560 s. The upper part shows seven targeted trajectories at different speeds and the lower pe tows the -• • • <lng velocity proliles (vertical calibration: 1 m/s).

28 P. Morasso separate the three segments of the hand path.) Remarkably, the peaks in the curvature profile temporally correspond with the speed valleys.

Moreover, similar findings can be shown also for three-dimensional movements (figure 5.2). The segmented appearance of curved paths and Its correlation with the structure of the speed were also found for different kinds of geometric constraints added to the main targeting task: (i) when subjects attempted to mimic constant curvature guides (see figure 5.3), (ii) when they tried to reach a target avoiding an interposed obstacle, or (HI) when the position of the target was changed during the execution of the targeted movement (see figure 5.4). Another common finding was that the modification of the speed profile (from straight to curved movements) was consistently accompanied by an increase of the movement duration, independent of the path length. A segmentation of the speed profile can also be observed in targeted movements for other types of constraints: (i) when the subject approaches the target in a time deliberately longer than the self - paced movement duration (usually around half a second; see also figure 5.5) [Morasso and Mussa Ivaldi 1963], and (ii) when the required accuracy (the ratio between movement length and target size) is great enough. In particular, the latter finding must be compared with the classic observation of Fitts [1954] that the duration of movements to a small target increases with the movement accuracy. The experimental findings are consistent with the notion that the basic motor primitive is a spatio-temporal unit with a rather fixed duration, with a stereotyped bell shaped velocity profile, and with a rather constant percent of accuracy. Within such a framework, the segmentation of the velocity profile in the previously mentioned paradigms can be interpreted as the attempt by the subjects to achieve a constrained trajectory by means of a chain of primitive targeted movements. Another remarkable finding is that the spatio-temporal structure of targeted movements, whether constrained or unconstrained, is independent of visual feedback and of the initial posture of the body. (The previous figures were recorded with sitting subjects who had their wrist and torso restrained, but similar findings were also obtained for standing subjects with a free wrist.) 5.2. Hand Gestures and Handwriting In the constrained targeted movements considered In the previous section, the deviation from the spontaneous nearly straight trajectories comes about ss a result of some additional type of external constraint In hand gestures (including hand drawing) and in handwriting, the hand path is not in accordance with an external constraint but Is derived from some kind of representation of the "shape" that is imagined and/or planned. (We may call this an internal constraint.) Let us now consider some experimental data about hand gestures and handwriting

[Morasso and Mussa Ivaldi 1982). Figure 5.6-left shows hand gestures which produce trajectories with a varying degres of "roundness".

Trajectory Formation 29 [£_, 1A^ c 3 _r i" O g Figure 5.6. Left column: three closed trajectories ol different 'roundness". Right column: three trajectories with different types of inflection points. Sampling frequency: 50 Hz. Velocity profile (V: calibration 500 mm/s). Curvature profile (C: calibration .04 Wmm left .006 1/mm right). Time calibration: 2.6 s |Morasso and Mussa Ivatdi 1082]. Figure 5.6-right shows hand gestures whose trajectories exhibit different types of inflection points. Figure 5.7 shows the repetition of the same pattern shape, while keeping the same pattern size, and the effect of systematically varying the pattern size, while keeping its structure. Figure 5.8 shows a three-dimensional gesture, in which the plane of the trajectory ch . =: continuously (as indicated by the torsion profile). Finally, figure 5.9 shows two examples of handwritten patterns. The most important finding is that all the patterns share a global common structure. In particular let us point out the following elements: (a) The curvature peaks are strongly correlated with the segmentation of the velocity profiles. (b) The curvature peaks are rather uniformly sequenced in time, independently of the sharpness of the trajectory bending (a finding which Is germane to the approximate isochrony of unconstrained targeted movements). (c) The sharper the trajectory bending, the greater the concurrent slowing down of the hand. (This fact is compatible with the simple vector addition of sequences of motor patterns with a bell - shaped velocity profile).

30 P. Morasso .\ ■' i : - o IVi Figure 5.7. Top: repetition ol the same pattern shape with a constant size. Bottom: repetition ol the same pattern shape with an Increasing pattern size. Sampling trequency: 50 Hz. Velocity profile (V: calibration 500 mm/s). Curvature profile (C: calibration .2 Mmm top, .008 Mmm bottom) IMorasso and Mussa Ivaldi 1982].

Trajectory Formation r\ .-••T-K ••*». s®m^ [WS* u ft ^™ Figure 5.8. Three-dimensional hand gestures. The little spheres (diameter 2 mm) represent samples ol the trajectory (sampling rate: 100 samples/s). V: tangential velocity (calibration 200 mrn/s). C: curvature (calibration 0.03 l/mm). S: torsion (calibration 0.03 l/mm). (x.y,z): components ol the binormal unit vector, i.e. the vector normal to the "osculating* plane. Duration ol the displayed movement: 8

s. {Morasso 1983].

32 P. Morasso (d) The above regularities are the necessary basis for producing the size invariant performance of figure 5.6- bottom. (e) Handwriting does not seem to be characterized by special spatio-temporal structures as far as the trajectory formation performance is concerned. A very interesting type of hand gesturing is the sign language which is used by deaf people. In particular, the American Sign Language has been studied in depth by Bellugi and coworkers from the point of view of the constraints on the language due to neuromotor structures. Recording the trajectories of the hands in space, they demonstrated that final position, not movement amplitude, is distinctive of the language [Poizner, Newklrk and Bellugi 1983]. Moreover, the overall kinematic structure of the trajectories Is consistent with all the other data on trajectory formation. Since hand gestures and handwriting produce complex signals (x-x(t), y-y(t), z-z(t) together with their derivatives), spectral analysis is an appropriate analysis tool to single out structural properties of the generation process. Figure 5.10 (Teulings and Maarse 1984] shows the average amplitude spectrum of the velocity time functions of a large set of handwriting movements. The spectrum shows a peak at about 5 Hz and vanishes beyond about 10 Hz. Consider now targeted movements: They have bell-shaped velocity profiles of about 400- 500 ms duration and therefore the corresponding spectra will also be roughly bell-shaped with a frequency band of about 2-2.5 Hz. The handwriting composite spectrum reveals an underlying periodic mechanism with an average period of about 200 ms (the mild sharpness of the peak suggests some fluctuation in the generation rate). In any case, such a peak can be related to the "Isochrony" of targeted movements. Furthermore, the energy of the spectrum between 5 and 10 Hz can be interpreted as being due to the "amplitude modulation" of the quasi - periodic C i c T Figure 5.0. Handwritten Panerns. Sampling frequency 50 Hz. Velocity profile (V: calibration 500 mm/a). Curvature prolile (C calibration .008 l/mm). Time calibration: 4.5 s (Morasso and Mussa Ivaldl 1982). U

Trajectory Formation 33 Figure 5.10. Averege amplitude spectrum of the velocity time functions of a large set of handwriting movements. The horizontal scale is in Hz [Teuiings and Maarse 1984). A. r-^A-^yjy Figure 5.11. Effect of low pass filtering on handwriting signal, (a): original written trace, (b): digitized trace (sampling frequency 200 Hz), (c to k): low pass filtering with cut-off frequencies of 10,7,6,5,4,3,2,1,.5 Hz, respectively [Teuiings and Thomassen 1979].

34 P. Morasso generation mechanism which tends to localize energy in the spectrum around the 5 Hz "carrier". Figure 5.11 [Teulings and Thomassen 1979] shows the result of filtering the Cartesian components of a piece of handwriting with a low pass filter of decreasing cutoff frequency. A 10 Hz cutoff frequency leaves the trajectory unaltered and a 5 Hz frequency still preserves the basic shape of the trajectory. It is only below 2.5 Hz that most of the trajectory shape is lost. All together, the indications provided by the frequency analysis of handwritten signals are compatible with the notion that complex trajectories are generated by a quasi periodic mechanism thai uses, as primitive elements, trajectory segments with spatio-temporal characteristics similar to the targeted movements. 5.3. Arm Body Coordination Arm body co-ordination is required when the hand trajectory exceeds the arm length. In Section 4.2 we discussed targeted movements, showing that the body moves, when necessary, in such a way that the trajectory of the shoulder "mimics" the trajectory of the hand. In the data shown, however, the distance of the targets was not so great as to force the subjects to step aside: In this case, it is possible to observe "pure" trajectory formation paradigms of the body and the arm. In this section we report on large size hand gestures and handwriting patterns that require stepping aside. These movements were recorded and analyzed with the same techniques of Section 4.2. Figure 5.12 shows some typical examples. The motor patterns of the hand are quite similar to those reported in the previous section. The trajectories of the shoulder are a more or less distorted version of the hand trajectories and are influenced by the concurrent stepping movement. Anyway, It is remarkable that the subjects, with apparent ease, can generate the same "extrinsic" movements in face of a widely different "intrinsic" implementation. This is another argument in favor of the central representation of hand trajectories. 6. MODELING TRAJECTORY FORMATION We have seen different motion paradigms, from targeting to hand gesturing, which all result in the generation of planar or spatial trajectories of varying degrees of complexity. The hypothesis that we formulate is that these paradigms of trajectory formation share a common generation mechanism [Morasso and Mussa Ivaldi 1982]. The model that we are going to discuss should fulfill the following specifications:

Trajectory Formation 35 Q i Figure 5.12. Arm/Body Co-ordination: Handwritten Patterns on a (Vertical) Blackboard. The top trajectory is drawn by the hand. The bottom trajectory Is drawn by the shoulder while moving the body (sampling frequency: 50 Hz). The extent of the trajectories Is about two meters. The two curves • i the tangential velocity ol the hand (the top one) and the tangential velocity ol the shoulder (the bottom one). Vertical calibration: 1.2 mis. Horizontal calibration: 1 s. (a) It is directly ex. = = • in terms of extrinsic instead of intrinsic co-ordinates. (b) It must produce smooth trajectories (without discontinuities at least up to the second time derivative5). (c) It must be powerful enough to produce any kind of shape (a requirement of com' - (d) It must produce the type of segmentation which characterizes human movements. (e) it must be • - -, i.e. it must be applicable in a uniform way to t • = : • movements as well as to handwriting. A promising approach for fulfilling these requirements Is provided by the family of methods which are used In computational geometry and computer graphics [Faux and Pratt 1979] for generating Parametric Composite Curves. A parametric •• --. tation of a curve, in general, is given as a vector (in 2-D or 3-D space) ex» ; • as a function of one independent variable. In our case, since we are considering movement trajectories, it is appropriate to use time as the 6 The first derivative determines the tangent to the curve and the second derivative determines Ihe radius ol curvature. Both are known to vary in a continuous way.

36 P. Morasso independent variable: (6.1) r - r(t) which corresponds to a couple of scalar functions in the case of planar movements x « x(t) (6.2) y - Wt) or to a triplet of functions In the spatial case x - x(t) (6.3) y - y(t) z - z(t). A parametric composite curve Is a curve in the form (1) which is obtained by means of composition, i.e. by joining simpler curve segments. However, before discussing the available methods, let us consider briefly some aspects of the differential geometry of curves. 6.1. Differential Geometry of Curves We are interested in "smooth" trajectories, which means that the functions in (2) or (3) must be continuous together with a sufficient number of their time derivatives. These derivatives determine the local, differential behavior of a trajectory, around a given time instant. For example, the tangent unit vector, t which specifies the direction, is proportional to the time derivative of r: (6.4) t - r' / |r'|. (The modulus of r' is the velocity V.) Two other significant vectors with regard to the local shape of a trajectory are (i) the "normal" vector n which points to the center of the "osculating circle" and (il) the "blnormal

vector" b which Is perpendicular to the plane of the osculating circle. The three unit vectors t, b, n are mutually perpendicular and they identify an "intrinsic" Cartesian frame of reference which moves and rotates along a trajectory: It moves in the direction of t, rotating around b as a function of the curvature C of the trajectory, and rotating around t as a function of the torsion or twisting T of the trajectory. For planar trajectories, b is fixed and consequently the torsion is zero. The movement of the Intrinsic frame along a trajectory Is characterized by a set of equations (the Frenet - Serret equations):

Trajectory Formation 37 f « +C Vn (6.5) n' » -C V t + T v b b1 - -T Vn where the curvature C and the torsion T must satisfy the following constraints: Cb.(f« r-yv3 <6-6> „ , t - r-. (r- x r-ytv6 cz) 6 The first of (6) can be interpreted by saying that the right - hand side of the formula computes a vector whose direction is b and whose amplitude is C. In the particular case of planar curves (b is constant) this equation reduces to: (6.7) C - (y" x' - y' x"VV3. The curvature and the torsion, together with the instantaneous velocity, give a complete characterization of the curve. In other words, the three time functions V - V(t) (6.6) C - C(t) T - T(t) in the three - dimensional case, and the two time functions V - V(t) (6.9) C - C(t) in the planar case, are an "intrinsic" representation of a trajectory, which is an elternative to the "Cartesian" representation (1). (It is worth noting that the intrinsic representation does not depand upon the system of reference.) (6) or (9) are derived from (1) by means of differential oparators and (1) is derived from (8) or (9) by means of integral operators. (8) or (9) are more useful from the point of view of analysis and (1) from the point of view of synthesis, i.e. of trajectory formation. 6.2. Parametric Composite Curves A parametric composite curve is one in which a chain of curved segments has been joined in a smooth way. Therefore, we have to detail three topics: (i) segmentation, (ii) nature of segments, and (iii) composition. With regard to segmentation, the independent variable (the parameter) Is time when dealing with trajectory formation. The composition mechanism implies the time axis is made discrete as a

38 P. Morasso sequence of time Instants ("knots" or "breakpoints" in the terminology of spline functions). These mark the subdivision of the overall curve into composing segments. Therefore, one of the reasons for choosing parametric composite curves as possible models of trajectory formation is that they imply "segmentation" in a natural way. Splines The kind of curved

segments that are mostly used in computer graphics are polynomial functions of time because they have attractive mathematical properties. (It is easy to compute derlvetives and the family of polynomial curves of a given order is a linear vector space.) However, this does not have any special biological meaning: Any (neural) process which might behave similarly as regards shape and timing would be computationally equivalent. Therefore, the rationale of the following detailed treatment of spline functions is just to provide an example of a feasible computational mechanism. In order to join In a smooth way two consecutive segments, it is necessary that at the "breakpoints" the final values of a number of time derivatives of the previous segment coincide with the initial values of the corresponding derivatives of the following segment. For example, in the case of cubic segments the smoothest composition is obtained when continuity is guaranteed up to the second time derivative (which means also continuity of curvature, according to (7)). The piecewise polynomial curves which provide this type of smoothness are known as "spline" curves (cubic splines when the segments are cubic polynomial functions of time). Algorithms exist for computing the coefficients of a spline curve, that is, the coefficients of each polynomial segment, on the basis of a set of interpolation points and the choice of boundary conditions. In our case, the natural boundary condition is that the first derivative must be null at the beginning and the end. Summing up, a spline curve is represented as follows: s.ft) L< t < U Sjjfl) U< t < £ (6.10) s - s(t) - * * J sk(tj tk<t<tk + 1 where t, are the breakpoints, and the polynomial segments s, may be represented as (6.11) Sj(t) - aj+ fcJ(t-tj) + Cj(t-tj)2A2+ dj(t-tj)3/6. The vectors ai,b,,Ci,d| are the coefficients which identify a cubic segment and which must be computed, when forming a trajectory, taking Into account the Interpolation points and the continuity constraints at the breakpoints. The algorithms which perform such computations are not Incremental: They require the advance knowledge of all the interpolation points and therefore are not well suited to carry out a * < > denotes the scaler product ot two vectors; <x> denotes the vector product.

 trajectory formation task. This drawback can be overcome by using a special kind of spline function - the B-spline. B- Splines The B - spline function is a spline which is null before a given breakpoint, is non - zero over the shortest time span and then goes back to zero. It turns out that the time span extends over a number of consecutive breakpoints equal to the polynomial order plus one (figure 6.1). Also the function is smoothly bell -

shaped. For a sequence of m breakpoints there are m-4 cubic B-spline functions and for each time instant there are 4 non - null cubic B - splines7. In order to maintain the same degree of overlapping on all the time spans, the above definition of cubic B-splines is completed with three additional couples of cubic B- splines: two are defined on three time spans, two are defined on two time spans, and two on one time span (see figure 6.1). Therefore, for a set of m breakpoints, the extended set of cubic B-splines contains m-4+6«m+2 functions. It can be shown that this extension is equivalent to adding three breakpoints coincident with the initial one and three breakpoints coincident with the final one. While the "internal" B-splines are maximally continuous at both ends, the "boundary" B- splines exhibit an increasing degree of discontinuity at one end (the left end for the initial B- splines and the right end for the final B - splines). The purpose of the extension described above is to have 4 cubic B- splines active for each time span. Furthermore, the amplitude of the B- spline functions is normalized in such a way that the sum of all the active B - splines for each time instant is identically equal to one: (6.1a) Sunij Bj(t) - 1 t^K^ j--a m-1 8 The practical importance of B-spline functions is due to the property [Curry and Schoen- berg 1966] that any spline function of a given order, defined on a given set of breakpoints, can be expressed as a weighted sum of the B - spline functions of the same order, defined on the same extended set of breakpoints: (6.13) s « s(t) - Sum, c, Bj(t) j--a m-1 The coefficients of the B - spline expansion are not interpolation values (except for the initial and the final one). They are guiding values which "attract" the shape of the curve maximally when time reaches the breakpoints where the B - splines reach their peak. Another Important property 7 This is true lor all lima spans except the initial three and the linal thru. In the lirst time span there is only one B-spline, in Ihe second there are two and in the third there are three. The order is inverted at the other end ot the interval. 8 In order to obtain (12) the B-spline lunctions are normalized in such a wey that their time Integral is equal to the lima interval in which they are non zero divided by the order ot the B-spline. The limits on the] index depend on the convention chosen lor numbering the B splines. The convention adopted is that the B- spline Index coincides with the Index ot the initial breakpoint lor that spline. The initial value ol j is -2 because the lirst breakpoint is multiple (-2, -1, 0, 1). 8 Since spline functions of a given order, lor a given set ot breakpoints, deline a linear vector space, the B- splines are a base lor such a space.

40 P. Morasso Figure 6.1. Cubic B-spline Functions. A: standard cubic B-spline (Its support is limited to four time intervals; note the convention of using the same index to indicate a B- spline and Its Initial breakpoint). B: standard cubic B- splines defined on a set of 6 breakpoints. C: extended set ol B-splines on the same set of breakpoints, that takes inlo account the border effects (note that this kind of extension guarantees that four B-splines are active for each time interval and that their sum is identically equal to one). of B-splines, with regard to the guiding points, is that for each time instant the shape is constrained to stay inside the "convex hull" of the guiding points whose corresponding B-splines are active. This guarantees that B-spline representations are "well behaved". From the computational point of view, it is useful to note a recursive relation [Cox 1973, De Boor 1972] that allows the computation of B- splines of a given order In terms of B- splines of a smaller order (i.e. cubic splines in terms of parabolic splines, parabolic splines in terms of linear splines, linear splines in terms of constant splines). This relation is currently implemented in most B spline calculations. B-spline Representation In the framework of a B-spline representation, a trajectory is formed by choosing the guiding points. A significant characteristic of the representation Is that it Is local (a modification of one of the guiding points affects the shape only in the time spans where the corresponding B - spline is non zero) and incremental, i.e. the trajectory can be generated "on-line" specifying new guiding points or changing pre - planned sequences of guiding points.

Trajectory Formation 41 The expansion (13) is directly applicable to spline functions. If we wish to apply it to parametric spline curves the only difference is that the guiding values are not scalars but are vectors (in two or three dimensions, depending on whether we are In = ; : • in planar or spatial curves): (6.14) r = rfl) « Sum, p. B,(t) j» -2,...m-1 where the p.'s are the guiding points of the trajectory. In the B-spline ■ • - ■ ntation of a trajectory these points play an absolute role, i.e. they specify attracting points in space which Influence the shape ol the trajectory. In the case of cubic splines, for example, the relation (14) specifies that at each time instant only four of these points attract simultaneously the trajectory and at each breakpoint there is a discrete switching of the four-points set (see fig. 6.2). S-spline • ■ However, it is possible to conceive the trajectory formation • • ►= ■ - In a complementary relative way, i.e. in terms of directions related to couples of consecutive guiding points. This approach leads to the definition of another set of basic spline functions: the S - splines [• = • and Mussa Ivaldi 1982]. While B-splines are bell- < • piece-wise polynomial functions, the S- apllnes are step- < < piece-wise polynomial functions and are formally defined as follows: % "i ". •» \ S. \ *r Figure 6.2. B-spline/S-spline Representation, intluence of guiding points/guiding vectors on the trajectory. The arrows on the left Indicate the portion ol the trajectory which is generated by the three consecutive guiding vectors (G3- >G4, G4- >G5. G5- >G6) which are active In the interval B3-B4. The arrows on the right indicate the adjacent portion ol trajectory generated by the three guiding points active during the Interval B4-B5.

42 P. Morasso (6.15) S|(t) « Sum, B.(t) i« -1... m -1; j«i +1 ,...m -1 It Is easy to show by using (12) that - 0 t < t. (6.16) S.(t) = 1 'M + 3 The above means that the active part of cubic S-spilnes is only three time spans (instead of four) and from this it Is possible to conclude that at any time instant only three S-splines are eimultaneously active (see figure 6.3). Spline functions (and spline curves) can be ■■>■■■ nted as an expansion In terma of S- splines in a similar way to the expansion (14) which uses B- splines: (6.17) r - r(t) - p_2 + Sumj (Pj - p._1)Si(t) j- -I,...m-1 The equivalence between (14) and (17) can be proved simply by substituting (15) In (17), while remembering (12). The S-spline representation is "relative" (instead of "absolute") because the trajectory formation ■ • :; • implied by (17) is characterized by guiding vectors (inatead of guiding points). Specifically, (17) can be interpreted as the vector addition of primitive straight movements ("strokes" in the motor terminology) directed along the guiding vectors: Each stroke is generated with a bell-shaped velocity profile (the time derivatives of S-splines are bell-shaped) and with a time shift among consecutive strokes characterized by the fact that at each time instant three of them are simultaneously active. From the point of view of trajectory formation, the relative point of view Is more attractive, because it results in more general motor formulations. •1 ** SI M n T» TJ T4 T> Tt Figure 6.3. B-splines and S- splines Defined over a Set ol Eight Breakpoints.

Trajectory Formation 43 6.3. Trajectory Coding A "smooth" trajectory (or a curve in general) is naturally segmented by the points of minimum radius of curvature (or curvature peaks). The segments delimited by two of such consecutive points, "segmentation points", have a stereotyped shape; they are either C-shaped if the curvature has the same sign at both ends or S-shaped, in the opposite case. In order to fit an experimental trajectory, it is possible to fit each segment after the segmentation points have been estimated: the shape of the curved segments can be accurately reproduced if we know the local behavior of the trajectory at the segmentation points10. Specifically, we can estimate a certain number of time derivatives at these points (e.g. the zero, first, and second order derivatives) and then we can compute a B- spline or S-spline representation that satisfies such constraints. In order to organize the computation, let us make the following hypotheses: 1. There are M segmentation points, to which we must add the two boundary points (M+2 points). 2. At each segmentation/boundary point we specify the values of the derivatives of order 0,1,2. We end up with (M + 2)3 constraints for each Cartesian component of our curve fitting problem: (6.18) (rk. r"k, r"k) at t=tk for k-l,...M + 2. (Note that for k = i and k«M + 2, the first derivative must be null.) If we now try to fit the overall trajectory with the B- spline representation (14) or the S-spline notation (17), we must guarantee that the number of unknowns (the guiding points or, equivalently, the guiding strokes) is equal to the number of constraints (specified by (18)). This condition requires that the number m of guiding points is equal to (M + 2)3. The overall balance among guiding points, guiding strokes, breakpoints, and segmentation points is summarized in Table I. TABLE I n < order of the polynomial segments > « 4 m < number of breakpoints > N < number of guiding points > 'm + 2 L<number of guiding strokes> « N-1 M < number of segmentation points > is such that N-(M + 2)3 In order to fit the recorded/segmented trajectory with the model (14) or (17) it is still necessary to identify the m breakpoints. The first and the Isst must coincide with the initial and final time instants of the trajectory. The others, taking into account the fact that N«(M + 2)3, can be arranged as follows11: m The rationale ol the statement comes Iran the Taylor series expansion ot a lunction In the neighborhood ot a point.

44 P. Morasso TABLE I h ui breakpoints l2 h J4 l5 l6 J7 •• u2 u3... segmentation points 'r UM- yt The fitting problem can then be expressed as the following set of linear equations: Surrij p, B (tk) - rk (6.19) Sum, Pj d(B, (tk)Vdt - r'k Sum| P| d2B| (tkVdt2 - r"k i=-2 m-1 k-1 M + 2 in the unknown vectors p/s. The solution can be found by solving separately the two systems corresponding to the two Cartesian components of each vector. Figure 6.4 shows an example of a curve fitted in this way. However, even if this kind of representation allows an accurate reproduction of the recorded trajectory, it is

not yet a satisfactory model of trajectory formation for at least two reasons: 1. It is limited to straight strokes. 2. It requires that three strokes are always active simultaneously. On the contrary, there is ground to think that curved strokes, as well as straight strokes, are primitive movements in the motor repertoire and that not more than two strokes are simultaneously active. In order to meet these requirements, it is possible to rearrange the representation given by (14) or (17) by observing that, taking apart the initial and final strokes from the set of L linear guiding strokes, we obtain a number of strokes that is a multiple of 3: (6.20) L - 3(M + 1)+1 + 1. Each triplet of linear strokes generates a curved segment (6.21) c_stroke|(t) - Pj_ 1Sj_ n(t) + p,S,(t) + Pj + 1Si + 1(t) with a bell - shaped velocity profile. We may call it curved stroke (see figure 6.5). We can then add two new entries to table I: 11 The regular arrangement ol table II oen be explained as tallows: (I) For M + 2 segmentation points there are M + 2-l-(M + 1) segmentation intervals, (ii) For m breakpoints there are (m-1) Interpolation intervals. (Hi) The constraint N-(M + 2)3, since N-m + 2, can also be expressed as m + 2»(M + 2)3. (iv) Rearranging the terms in (iii). (m- l) + 3-(M +1)3 + 3, comes (m- t)-3(M +1). This is the result which determines the regularity of the table. Note thai such a regularity depends on the number ot constraints that we expressed at each segmentation point. For other configurations, in general, we end up with more complex correspondence schemes.

Trajectory Formation s s K i / / <;- A" iS\ "rH'tV Figure 6.4. Trajectory Fitting with a S-spline R ■ = ntation. (B1 BIO): breakpoints. (SI,...Sit): S- splines. (G1 G11): guiding points. The two small circles identify the segmentation points. \ ***+m»* Figure 6.5. Curved Strokes Obtained as Combination ol Three Linear Strokes. A- curved stroke is not necessarily symmetric. It can be either C- shaped or S- shaped. Its duration is live intervals. Note that the velocity profile is bell shaped. K < number of curved strokes > =(M + 1) NT < number of strokes of the generalized representation > - K + 2. Linear strokes are identified by two parameters (direction and length) whereas curved strokes are identified by six parameters. (Two have the same meaning, two are related to the 'roundness" and two to the "skewedness" of the stroke.)

46 P. Morasso The S- spline representation (17) can then be re-written as a generalized curved stroke representation: (6.82) r - r(t) » Sum. c_strokej(t) + p_inWal J«1,...K This says that, in general, a complex trajectory can be formed

by a process that generates sequences of curved segments (with a stereotyped velocity profile and with such a timing that no more than two strokes are simultaneously active) and composes them additively. It must be stressed that representation (22) is formally equivalent to (14) or (17), but it better captures the structure of the human trajectory formation process. 6.4. Cursive Script Coding Handwriting is a special type of trajectory formation paradigm which is characterized by the fact that the generated trajectories are not constrained in a 'physical" way but In a more abstract "propositions!" way. For example, the "picking up an object" implies to plan a trajectory that drives the hand to a specific position in space with an appropriate orientation determined by the shape and location of the object, whereas "cursive script" Is the motor translation of a symbolism. As a consequence, in cursive script analysis it is significant to consider special trajectory formation concepts which are meaninglesa in general. For example, since the Greek-Roman families of languages are characterized by a left-to-right organization, it Is natural to consider the concept of a baseline as the Ideal line which marks the progress of letters Into words and of words Into sentences. The baseline, according to the acknowledged cursive script conventions, is assumed to be straight and horizontal, but the use of cursive script as a communication medium requires a large tolerance on the actual shape and orientation of the baseline, provided it is smooth and slowly changing, at least at the word level. The baseline provides one ideal axis for the trajectory formation task associated with handwriting. The second ideal axis is orthogonal to it, so Identifying a natural system of reference. Ideal cursive script shapes12 are defined in the natural system or reference, together with "attributes" of the composing elementary shapes. For example, In the following definitions "a long straight stroke down to the baseline* or "a short narrow loop above the baseline" "straight stroke" and "narrow loop" are parametrized primitive shapes oriented In the natural reference. "Short" and "long" are size descriptors which have a relative meaning for the overall script. "Above" and "down* are positional descriptors with respect to the natural system of reference. The same reasons of robustness which force the community of readers/writers to tolerate a variability In the baseline are good enough for accepting a non - orthogonal system of reference, where the slant may be due either to mechanics (the use of two oblique degrees of freedom) or to cognitive aspects (the choice of a particular writing style) or to both. It is reasonable to think 12 Each "Ideal" shape admits a number ot valid variations - the "allographs" - which are a sort of 'motor svnonyms".

Trajectory Formation 47 that writers tend to adopt writing postures in such a way that the resulting functional degrees of freedom match the desired slant. Summing up, in modeling handwriting as a trajectory formation task, it Is necessary to separate two distinct aspects: 1. Global aspects which have to do with the natural reference system and are captured by the notions of baseline, slant, size, and 2. structural aspects, which are captured by the (arbitrary/individual) choice of primitive formal elements used to construct allographs. Therefore, cursive script coding, which is a prerequisite for cursive script understanding, requires separation of the two types of aspects. From the point of view of analysis, thie meana to identify two main tasks: (a) normalization, (b) segmentation. Normalization of Cursive Script Four types of normalization can be distinguished: (i) time scaling, (il) baseline compensation, (III) slant compensation, (iv) size scaling. The need of time scaling comes from the fact that different people have different writing speeds and the same person can write at different speeds. What remains constant is the shape of the frequency spectrum of handwriting as a signal, which is characterized by a dominant peak around 5 Hz [Maarse and Teulings 1984]. Changing the writing spaed displaces the peak frequency or, expressed differently, varies the overall number of strokes per second. Time normalization can then be defined as the problem of scaling the time axis (in practical terms, resampling the signals) In such a way that a standard writing speed Is obtained, which can be measured either In terms of the peak frequency or of the stroke rate. After time normalization has been performed, measures of durations, delays, etc., which are important In cursive script processing, become relative. The specification of the particular writing speed Is stored In a single scaling factor. The second type of normalization involves the baseline. The baseline may be defined as a smooth line which joins (or is close to) the bottom part of the strokes, except for some of them (e.g. the long strokes in "g and ("). The notion of baseline Involves a "selection" and this rules out the possibility of estimating It by low pass filtering the handwriting signals with a very tow cutoff frequency (at least ten times smaller than the peak frequency), because such a filtering le a linear operator which does not select specific features. Nonetheless, the curve which is obtained by low pass filtering is not far from the baseline because the "exception strokes" are not statistically dominant. It can be used at least to give a rough indication of right/toft, and up/down. Once that is done, it is possible to carry out the selective process by: 1. Trading the strokee as the handwriting segments limited by adjacent points of peak curvature (segmentation points), and

 2. selecting the segmentation points where a down-going stroke is followed by an up-going stroke and at least one of them is a "small" stroke. It is clear that In this algorithm there are parameters that require auxiliary procedures: "Up/down" requires a rough estimate of the baseline and "small/big" requires a statistical analysis of the distribution of stroke lengths. When the appropriate segmentation points have been selected, it is possible to fit them with a smooth curve (6.33) b - b(t) which is an estimate of the underlying baseline. The normalization of the baseline can be obtained by "deforming" the handwritten trace in suoh a way that the resulting baseline is straight and horizontal. The deformation should compensate the rotation and translation which are due to the curvedness of the baseline. Specifically, we can derive from (23) the three time functions delta_x « delta_x(t) (6.24) delta_y = delta_y(t) theta - theta(t) which identify (i) the displacement vector between a point on the baseline and the corresponding point on the handwritten curve, and (ii) the rotation angle between the tangent to the baseline and the horizontal13. Finally, the baseline curvedness and tilt are compensated by transforming each point of the original trajectory, referred to its corresponding baseline point, with the following homogeneous transformation matrix cos theta(t) -sin theta(t) delta_x(t) (6.25) sin theta(t) cos theta(t) delta_y(t) 0 0 1 which performs the required rotation/translation (for details about homogeneous matrices see the appendix). The third type of normalization Involves the slant. As in the case of the baseline, two problems need to be solved: (i) (slant) estimation, and (ii) (slant) compensation. Slant may be defined as the average orientation with respect to the baseline of the handwritten trace at specific points. For example, we may select only the points of peak curvature or the points of peak velocity or, among them, only those that correspond to up-going or down-going strokes. In any case, after the selection criterion has been defined (the range of choices is caused by an arbitrary margin intrinsic in the notion of slant) and the average orientation at the selected points has been computed, slant compensation requires a "shear transformation" which makes the selected average orientation either zero or ninety degrees depending on 13 The correspondence can be established on the basis ol the curvilinear co-ordinate, hi other words, the time course I • l(t) ol the curvilinear coordinate is computed integrating over time the spaed function and then, far each time instant tt, Ihe position on the ideal baseline is taken to

hava a distance from the origin equal to Iftt).

Trajectory Formation 49 the selected orrterion. The compensation can be implemented by means of a homogeneous transformation matrix 11 0 01 (6.26) -tan(alpha) 1 0 I 0 0 11 that transforms an oblique co-ordinate system Into an orthogonal one. (Alpha Is the angle between the two y co-ordinate axes.) The fourth type of normalization involves size. This means two things. One is the relative scaling between the horizontal and the vertical components of the script in such a way that some average shape criterion is met (e.g. normalized average "roundness" which is known to reliably classify male and female writers). The other Is the global scaling In order to obtain some standard size parameter (e.g. the average stroke length). Segmentation of cursive script After the normalization procedures have been applied to an Individual cursive script, we can store individual global parameters (time, size, baseline, slant) and we can apply to the normalized traoe the segmentation and coding concepts already discussed for trajectories in general. In particular, we can use the same fitting procedure expressed by (19) which constructs the curved stroke representation (22). As already remarked, these kinds of strokes are characterized by six parameters. However, it is worth noting that, at least for "regular" handwriting [Morasso and Mussa Ivaldi 1982] it Is possible to fit the data with circular strokes which have only three parameters (length, orientation, and radius of curvature) as it Is shown In figure 6.6. In any case, the final result of the segmentation phase is a representation of the script as a chain of strokes characterized by shape descriptors. Such a script notation is the starting point for structural shape analysis. An objective which is left to the future Is to build allograph databases to be used in the automatic recognition of cursive script. 7. DIRECT AND INVERSE KINEMATICS To a first approximation, the human body can be modeled, from the motor point of view, as a system of rigid bodies. In the global system we can single out kinematic chains (arms, legs, fingers etc.) which consist of sequences of links and joints, ordered from a base joint to an end effector. Each joint may have one or more degrees-of-freedom (i.e. Independent rotations q1,q2,...).

50 P. Morasso Figure 6.6. Fitting handwriting trajectories with circular strokes. D: experimental data. S: (circular) stroke representation. M: resulting trajectory of the model. Bottom figure: velocity prolite ol the experimental data and of the model (vertical calibration: 30 cm/s. sampling frequency: 50 Hz;

sampling window: 5 s). Modeling kinematic ohains is common to robotics and biomechanics. The kinematics problem consists of two subproblems: direct and inverse kinematics. The direct problem is to find the position and orientation of the end effector of a kinematic chain with regard to a reference coordinate system given the joint variable vector, q-(qi,q2 qn), where n is the number of degrees - of - freedom (dof). Also required are the various link parameters. The Inverse problem (or arm solution) is to find the joint variable vector q for positioning the end-effector of the arm at the desired position with the desired orientation, given the position and orientation of the end effector with respect to the reference co-ordinate system and the various geometric link parameters. In both cases, in order to describe the relative movements of each link with respect to the previous one in the chain, or the absolute movement with respect to the environment, It Is convenient to associate a local system ofco- ordinates to each link, as It Is shown in figure 7.1. When a link rotates around Its joint all the points of the link change their oo-ordinates with respect to the Cartesian frame of the preceding link, while obviously maintaining constant their co - ordinates with respect to the local frame. Such a transformation can be expressed In matrix form by using 4x4 homogeneous matrices and 4 - dimensional homogeneous vectors. (See the appendix.) For example, if

Trajectory Formation 51 Figure 7.1 Schematization of an Anthropomorphic Arm with Different Systems of Reference. h1«[x1,yl,z1,1]T 14 is the homogeneous vector which stores the co - ordinates of a point of a linkage in the linkage frame F1, then the homogeneous vector which identifies the same point in another frame F2 is h2 -2H1 hi where 2H1 is a homogeneous matrix that expresses the rotation/translation of F2 with respect to F1. H Is a function of the joint variables (e.g. three variables for the shoulder, one for the elbow, etc.). The structure of H is shown in the appendix. What is important is that the joint transformations can be cascaded because the output of one is the input of the other and this means that the transformation due to a kinematic chain can be expressed by the matrix multiplication of the corresponding homogeneous matrices. Specifically, in the case of the anthropomorphic arm of figure 7.1 we can write: (7.1) hO-^H^H^Hgrfl where h3 Is a point of the hand referring to its co-ordinate system, hO Is the same point referring to the environment or base system, and the three matrices express the rotation of the shoulder, elbow, and wrist joints, respectively. Since the homogeneous matrices are known functions of the joint variables, then the previous relation is the mathematical formulation of the direct kinematics problem. 14 The notation vT is used to denote the transpose vector, where transposing mans to exchange rows with columns.

 The inverse problem is much more complicated and delicate: (a) The equations which need to be inverted are non - linear. (b) Many different arm configurations correspond in general to the same configuration of the hand. (The inverse equations admit multiple solutions.) (c) The possible configurations become infinite when the number of degrees of freedom exceeds six15. A closed form solution such as that of the direct kinematic problem does not exist in general. However, solutions have been found for specific types of arms. For example, Lee [1983] reviews several methods used for 6 dof industrial manipulators and Benati, Morasso, Tagliasco [1982] show a method for 7 dof arms with an anthropomorphic structure. The latter method, in particular, gives up the general solution and solves the problem In an incremental way, i.e. sample after sample along a trajectory. For each time slot, the algorithm consists of two steps: 1. Rotation of the elbow which adjusts the wrist - shoulder length to the required value. 2. Rotation of the shoulder which carries the wrist onto the required location. (From this rotation it Is possible to derive also the wrist rotation, taking into account the "coherence" of the arm.) The two rotations can be computed by means of simple vector algebraic calculations. Moreover, the redundance of the arm can be taken into account by adding, at each step of the algorithm, another rotation: 3. Combined shoulder - wrist rotation around an axis which passes through both joints. and by adjusting such rotation in order to match some additional constraint (e.g. avoiding an obstacle) or to keep some performance index close to its optimum. The direct kinematics plays an important role in proprioception and in haptic perception. The primary and secondary afferents from the muscle spindle as well as the joint receptors provide indeed abundant Information about the relative movements of the joints. But what is essential for purposeful action is to know the position in space of the different parts of the body or the position in space of tactile information. Inverse kinematics, on the other hand, as was argumented in the previous sections, is the basic ingredient for the representation and production of space oriented motor programs. 8. DIRECT AND INVERSE DYNAMICS The dynamics of a kinematic chain is due to the masses of the links. (If the links had negle- gible masses, then the movement of a kinematic chain would be only a kinematic problem.) The 16 The configuration of the hand in space is identified by six variables - three orientation and three translations. Arms which heva more that six dot are called redundant. The human arm, even neglecting the translations ot the shoulder, has at least seven dot.

Trajectory Formation 53 presence of masses implies that the arm can store kinetic and potential energy: Its movement is then dominated by differential equations. The dynamic differential equations can be expressed in different ways, which have different computational characteristics and therefore have received a great deal of attention in the robotic literature. The classic formulations of Lagrange, Newton - Euler, and D'Alambert have been adapted to manipulator arms [Orin et al. 1979, Benati et al. 1980, Luh et al. 1980, Hollerbsch 1980, Paul 1981, Silver 19831. The Lagrange formulation is particularly compact and reveals quite clearly the type of non-linearities and cross couplings that make the dynamics of kinematic chains complex. The starting point is to select a set of free variables q1,q2,...qn (They are also called generalized co-ordinates in the Lagrangian terminology.) which correspond to the n dof of a kinematic chain16. Then, the dynamic equations of a multi-body system, according to the Lagrange formulation, can be written as follows: (8.1) Q|» d(DL / Dq'.ydt - DL / Dq| i» 1,2,...n where L « Lagrangian function = Kinetic energy (K) - Potential energy (P) q'l - generalized velocities a « generalized co-ordinates Qj « generalized forces n « number of dof's D « symbol of partial derivative For a kinematic chain, we can take into account two facts: 1. The kinetic energy is a quadratic function of the generalized velocities K - 1/2 (q')T Kq) q'. (The elements of the matrix I have the dimensions of moments of inertia.) 2. The potential energy depands only on the generalized co - ordinates P - P(q). If we apply these relations to (8.1), we end up with the following dynamic equations of kinematic chains: (8.2) Q - Kq) q" + Ctq.q1) + G(q) where 16 The choice of Irse variables Is not unique. For example, If we consider the shoulder |olnt we have 3 dot but we have different possibilities tor the free variables: Euler angles, Bryant angles. Rodrigues parameters etc.

54 P. Morasso q « vector of generalized variables q' « vector of generalized velocities q" m vector of generalized accelerations I « matrix of inertia C « (generalized) forces of Coriolis Q « (generalized) forces of gravity Q «

(generalized) forces of the actuators These equations express mathematically the Inverse dynamic problem which can be stated as follows: Calculate the time course of the actuator forces which are necessary to drive an arm along a prescribed trajectory with a prescribed law of motion. The equations show that the actuator forces must balance three types of resisting forces: 1. forces due to the accelerations (note that the inertia expressed by I is a function of the arm configuration q); 2. forces due to Coriolis effects, i.e. forces that depend on products of velocities, with coefficients which depend on the configuration; 3. forces due to gravity that depend only on the configuration. The combined effect of these forces is characterized by (I) non - linearity, (ii) dependence on the configuration, and (iil) cross-coupling among the degrees of freedom. The consequence, for example, is that it is not possible to generate appropriate motor commands for a synergic group of muscles without taking into account the disturbances due to the movements of the other dof. The computation of the actuator forces given by (8.2) Is relevant for the motor controller and different schemes have been proposed to perform It efficiently. Although the detailed analysis of the algorithms is outside the scope of the book, we deem It important to emphasize that efficient algorithms do exist which solve In a feedforward way the Inverse dynamic problem, i.e. straightforward computational modules able to predict the time course of the actuator forces necessary for compensating the non-linearities and the cross-couplings due to dynamics. That is important because kinematics and geometric planning are logically on a higher level. The appropriate level, we think, at which human movement can be understood and one of the reasons that support this assumption is the availability of feasible computational "slaves" which free the "master planner" from the complexities of dynamics. We end this section and the chapter briefly mentioning the complementary dynamic problem - the direct dynamic problem. This problem consists of computing In the phase or state space17, the arm trajectory which Is determined by an assigned vector of actuator forces. It is equivalent to solving (8.2) as a set of differential equations, with Q as the input driving vector function. This problem is more interesting from the point of view of simulation than from that of motor control. (In the human case, the integration of the differential equations Is performed Implicitly by the human body.) In the field of simulation, standard techniques exist for Integrating numerically systems of ordinary differential equations. They assume a formulation of the dynamic equations of " II q is the n-dimensional vector ol Iree variables, the state vector is x-(W) which has 2n dimanelons. This is due to the lact thet the state of a mechanical system, i.e. ite memory, Is determined by the stored energy and therelore it depends on both configuration and speed.

Trajectory Formation 55 the form: x' - Kx.t)- Equations (8.2) can be expressed simply if we write q" as a function of everything else: q" -[Kq)l"1[Q(t)-C(q.q')-G(q)l from which we get HqM) r1[Q-C-G]

56 P. Morasso APPENDIX: Homogeneous Matrices A matrix is a linear operator that transforms a vector into another vector. For example, a 3x3 matrix transforms a 3-dimensional vector In another 3 - dimensional vector. Output_Vector < - Matrix < - lnput_Vector In general, the transformation changes both the direction and size of the vector. However, If the size remains constant, then the transformation is a pure rotation and the corresponding matrix is called a rotation matrix. Rotation matrices must satisfy an orthogonality constraint: R RT - I » From the geometrical point of view, the rotation of a vector can be considered also as a change of co-ordinates, from an original Cartesian frame to another frame which Is rotated with respect to the previous one. The rotation between the two frames Is perfectly known when we know the co-ordinatea of the unit vectors of one frame (I1j1,k1) projected on the other frame (I2j2,k2): 11.i2 M.j2 11.1(2 J1.I2 J1.J2 |1.k2 1B k1.l2 k1.j2 k1.k2 and, vice versa 12.11 I2.J1 I2.M J2.i1 J2.J1 J2.k1 k2.H k2.|1 k2.k1 It is not difficult to show that the nine numerical elements shown above In either the first or the second set, are just the elements of a rotation matrix. We obtain one set from the other by exchanging rows and columns. This fact expresses the orthogonality condition stated above. One of its consequences is that the nine elements of the matrix are not Independent but are functions of three free variables20. These variables can be chosen In Infinite ways. For example, In the case of the classic Euler angles, one frame is obtained from the other by means of three rotations: (i) A rotation around the z-axis (psi« angle of precession), (II) a rotation around the x-axis (theta«angle of nutation), and (Mi) a rotation around the z-axis (ft»angle of proper rotation). The structure of a rotation matrix In terms of the Euler angles Is then obtained as the product of three matrices which express the three primitive rotations (figure A.1). Another possibility, is to consider the (finite) rotation which allows to transform one frame Into another. This rotation " I Is the Identity metrlx, a matrix In which the elements of the principal diagonal are 1 and all the other*

an 0. TM* matrix performs an Identity rotation moaning the output vector is equal to the Input vector. M' nanofe the tranapoae matrix, I.e. a matrix which is obtained by exchanging rows with columns. " The co-ordinates of a vector with regard to a Cartesian frame of reference are the projections of the vector onto the threa co-ordinate axes and they can be computed by the scalar product* of tha vector with the unK vectors of the axes. 20 The free variables of a rotation matrix are three because the rows (or the column*) of the matrix mutt have unitary length (three constraints) and must be mutually perpendicular (three constraints).

Trajectory Formation cp sp 0 -sp 0 + cp 0 0 1 1 0 0 ct 0 st 0 -st ♦ ct cf -sf 0 sf +cf 0 0 0 1 -cp sf - sp ct cf +sp st stcf ct Cp Cf — Sp Ct Sf — \jy) si — 9y ui ui Top oi spcl + cpctsf -spsf + cpctcf -cpst st sf "•M "♦ cp ■ cos(psi) ct - cos(theta) cf - cos(fi) sp « sin(p8i) st « sin(theta) af - sin(fi) Figure A.1 Rotation Matrix: Euler Angles is characterized by a rotation axis (unit vector n-[x,y,z|) and by a rotation angle (alpha). The structure of the rotation matrix can be shown to be that of figure A.2 [Faux and Pratt 1979J21. Rotation matrices express co-ordinate transformations where the two Cartesian frames have coincident origins. If this Is not true, the transformation will include a translation In addition to a rotation. Homogeneous matrices allow to express both at the same time. These are 4x4 matrices with the following structure: . x ■y 000. 1 where R Is a 3x3 orthogonal submatrix and [x,y,z]' Is the translation vector from the origin of one frame to the other. The corresponding homogeneous vectors are 4-dimensional: 3 - dimensional vector v - [x,y,z]T 4 - dimensional vector h « [xw,yw,zw,w]T (w can have any value except 0). It is Immediate to show that, If hi Is the homogeneous vector x x (1 -c) + c y x (1 -c) - z s z x (1 -c) + y s x y (1 -c) + z s y y (1 -c) + c z y (1 -c) - x s x z (1 -c) - y s y z (1 -c) + x s z z (1 -c) + c c - cos(alpha) s « sin(alpha) [x,y,z]'« rotation axis Figure A.2 Rotation Matrix: Rnite Rotations. " tl le worth remembering also another set ol free parameter*: the Rodrtguee parameter* |BiMhop 1980]. They am the three components ol a pseudo-vector which is defined as v-tan(atpne/2)n.

58 P. Morasso that Identifies a point In a frame F1 and h2 Is the homogeneous vector that identifies the same point in F2, then the transformation from hi to h2 (or vioe versa) is Implemented by the homogeneous matrix whose rotation and translation components correspond to the rotation and translation between the

two frames. In a kinematic chain, cascading co-ordinate transformations Is implemented by matrix multiplication. In addition to rotation and translation, homogeneous matrices can be used to express other geometric transformations: (I) shear transformations, (II) changes of scale, (III) planar projections etc. [Foley and Van Dam 1962]. It is also possible to consider 2D geometry instead of 3D geometry: In this case we can define 3x3 homogeneous matrices and 3x1 homogeneous vectors with a very similar formalism.

PARTI Chapter 2 COMPLEX MOTOR PATTERNS: WALKING Luigi Baratto, Pietro Morasso and Renato Zaccaria CONTENTS

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasco (Editors)
COMPLEX MOTOR PATTERNS: WALKING Luigi Baratto, Pietro Morasso and Renato Zaccaria In generalizing the concepts on motor control discussed with regard to trajectory formation, we first contrast the direct and reverse operation of kinematic chains which represent the two fundamental modes of moving a skeletal structure. Then we attempt a formalization of a complex but usual motor skill - walking - in which the main different representational aspects are present. The goal is to stress concepts and mechanisms which are possibly common to many other skills, without focusing on details which are typical of just walking. Finally, we look at the same problems from the point of view of movement pathology in the case of lesions of the Central Nervous System which are supposed to determine "program disorders". 1. LIMBS: THE DIRECT FUNCTION AND THE REVERSE FUNCTION Humans, as many other animals, have four limbs. Two of them are specialized for manipulation whereas the other two mainly serve locomotion. In fact,

manipulation and locomotion are regarded as two largely different motor activities not only from the point of view of common sense but also in the academic research field where the literature on the subject is rather disjoint. On the contrary, in our opinion there is much to be gained by looking at both manipulation and (legged) locomotion as special paradigms of the same motor topic: co-ordinated control of a system of limbs. A "limb" is a kinematic chain, i.e. a sequence of articulated linkages which originate from a "reference body" (e.g. the pelvis) and terminate with an "end effector". A limb, for its mechanical structure, can serve two main functions: (a) a direct function, and (b) a reverse function. The direct function is essentially what we called, in the previous chapter, trajectory formation, i.e. the generation of arbitrary trajectories in space by means of the end effector. The reverse function is the generation of movements of the body in space while keeping the end effector of the limb fixed with respect to the environment. With regard to the lower limb, both functions can be found in walking. The swing phase is an example of the direct function, whereas the stance phase is an example of the reverse

function. Similarly, for the upper limb targeting or handwriting are examples of the direct function and the arm movements induced by keeping a handle are examples of the reverse function. The two types of functions can operate concurrently^, e.g. in using a screwdriver or in extinguishing a cigarette butt under the heel. The human ability to switch from one function to the other in controlling the movements of a limb, or even to mix the two functions at the same time Is a strong argument in favor of the limbs being primitive motor structures in addition to joints and muscles. Significant formulations of movement paradigms can be given either in terms of joint control or in terms of limb control, without any rigid hierarchy between them. Since the muscles are "relative" actuators, i.e. they transmit forces across joints, the global effect of a given muscular activation pattern depends on the configuration of the reaction forces of the environment. These forces counteract both the forces due to gravity and the forces due to the muscles. The geometry and mechanics are in general very complex, but in most cases it is possible to make hypotheses that drastically simplify the problem understanding and consequently, the possibility of controlling it. For example, when we are sitting and reaching for objects on a table, very complex force distribution patterns will be established but we know that if the chair is a regular chair (i.e. it is strong enough, it is not slippery, etc.) and the movements are not too brisk, then the body will remain about still and the action of the arm muscles will cause the hand to trace a predictable trajectory. That would not be the case, for example, in the situation of being suspended from a rope, or falling down in free space. The same muscular contractions of the arm would produce quite different trajectories. Therefore, when we speak of trajectory formation and perhaps think of it in terms of equilibrium points of set of muscles, we must assume that the environmental conditions are such that the cross - coupling between the limb and the rest of the body is insignificant. (Borrowing an expression familiar in electrical circuits, we may say that the body is put to "ground".) Similar concepts also apply to the control of a limb in the reverse way. The hypothesis that we must make is that the terminal part of the limb is effectively "grounded"2. Then the same muscle contractions that would cause a certain trajectory of the free limb in a "grounded" body will generate a complementary trajectory of the origin on the limb in the opposite direction. When modeling the movements of a humanoid (see the chapter on the simulation language NEM) the concept of the reverse function of a kinematic chain (i.e. a limb) Is very Important. Many common movements would be virtually impossible without the possibility of coordinating the direct and reverse control of limbs. Consider for example the simple movement of picking up something while holding a handle as it may happen when one is standing on a bus. The body and the picking hand move toward the target (according to the direct modality); the holding hand keeps the handle without obstacling the body; and the legs shift the body without moving the feet, both operating according to the reverse modality. Although muscle compliance is ' Concurrency, as stressed also in other chapiers. is a fundamental motor concept. It carries two main Ideas: (i) Computational parallelism, and (n) geometric composition ol the effects, which is implicitly performed by the "limbed" structure of the body. 2 A reliable connection to the ground (without such artificial constraints as strappings etc.) depends strongly on Motion. We all appreciate the extent of our implicit assumptions about the quality ot the ground when we act in a low friction environment.

 likely to play an important role in this particular paradigm as well as in all constrained movements, the overall co-ordination is the main motor problem. Specifically, co-ordinating the holding hand with the body, the body with the legs, and the picking hand with the body, makes sense only if the common, unitary function (getting the target) is the source representation from which all the movement patterns (through direct or reverse action of the limbs) are "computed". Let us now consider these concepts in greater detail with respect to a specific motor pattern: walking. 2. WALKING Bipedal human walking is a process of locomotion and a motor skill in which the body weight is alternatively supported by one foot then the other, with an intermediate phase of double support. As a result of this alternation, the center of gravity may be moved forwards, or backwards, or sideways with regard to a system of reference fixed for example in the pelvis. For any specific walking direction, in combination also with physical and geometrical characteristics of the ground, different walking patterns will emerge. Specifically, when moving forward on a horizontal, non-slippery surface ("level walking"), the pattern of movement is characterized as follows: One foot moves ahead of the other with the heel of the forward foot touching the ground before the toe of the opposite foot pushes off, and the arms and legs move synchronously in opposition. In certain conditions this pattern evolves to running which is a natural extension of the same basic skill. The distinguishing factor in running is a phase in which the body is propelled through space with no support from either leg. Uniform walking, without considering the initial and final phases, results in a cyclic pattern where all the relevant kinematic variables are characterized by periodic waveforms. For example, in level walking the center of mass oscillates quasi - sinusoidally on both the plane of progression and the horizontal plane, with the frequency of the former oscillation being twice the frequency of the latter. The kinematics and kinetics of this pattern has been studied extensively [Inman et al 1981] ss well as its physiology. In particular, it has been found that in many different species the basic periodic patterns of walking are preserved after decerebration, suggesting the hypothesis that the central pattern generator for walking consists of independent neural oscillators, each associated with a single limb joint. The corollary of this point of view, which may be literally right, is that purposive behavior (e.g. locomotion in a structured environment) is considered simply as a "modulation" or an "adaptation" of the neural oscillators. The problem with this point of view is that it focuses on one sspect of walking, periodicity, which is the most showy but is possibly only a byproduct of a structured process. To reduce the control of walking to the modulation of a set of coupled oscillators is the same as to reduce speech only to vowels, which are indeed periodic waveforms, or handwriting only to a modulation of an oscillatory pattern.

 On the contrary, walking is the result of many complex sensory - motor integrative actions [Qrillner 1975, 1961]. They are (i) the generation of an efferent plan, (ii) the continuous adaptation to expected and unexpected external conditions on the basis of afferent information [Roberts 1976, Thompson 1980], (iii) the comparison between the planned sensory - motor patterns and the actual ones, and (iv) the dynamic maintenance of equilibrium [Nashner 1960, 1985]. In other words, we think that a complex motor pattern such as walking can be understood only in a cognitive framework as the co-operation among several motor processes which produce periodic waveforms only in special conditions. The power of the representation should be its generality and its "cognitive penetrability" [Pylyshin 1984) to variations of the basic locomotor pattern such as non uniform walking, walking upstairs and downstairs, overcoming obstacles, etc. For example, in moving backwards, the first part of the swinging foot which touches the ground is the toe instead of the heel. In moving sideways there is a clear asymmetry between the movements of the two legs. When locomoting in a cluttered environment or when performing complex locomotion patterns, such as in dancing, the "superficial" structure may change widely but we think that it Is reasonable to assume that what we observe are just variations of the same locomotor skill.

2.1. A Formalization of Walking

A way of formalizing human motor skills is to build an ensemble of co-operating "motor actors" which concurrently operate on the same data structure - the humanoid skeleton. The actors are cognitive units, which capture some aspect of the global skill and implement the corresponding "computations" either in an explicit way or in an implicit - analogic way, with or without the contribution of sensory information. Parts of the model of the walking skill have been implemented in a simulation language, NEM, that is described in another chapter of the book. Since walking implies an alternation, although not necessarily a periodicity, it is natural to model each actor of the "walking skill" as a finite state automaton [Aleksander 1978) and the whole process as the co - ordinated activity of a group of automata. The notion of automaton is that of a mechanism characterized by a "set of internal states". The current state is a function of the "history" of the automaton, i.e. the stale is a form of memory. The behavior of an automaton depends on the state, that Is, the automaton may respond to the same stimulus in a different way, according to the state in which it is. A finite state automaton can have its behavior characterized by the specifications of what it is doing in each of its enumerated states and by the events which cause it to switch from one state to another. A "state transition diagram" is just a graphic way to represent the states and the state transitions. In particular, we consider the following set of automata: (a) a foot-path automaton, (b) a balance automaton, (c) a left-leg automaton, (d) a right-leg automaton. The path automaton expresses the goal of the skill (i.e. locomotion) and the basic method for attaining it (i.e. walking). In particular, it does two things: (i) It anticipates the sequence of

"foot prints", one after the other, and then (ii) it "moves" the pelvis in the corresponding direction. The incremental planning of the foot path expresses in general terms the abstract geometric structure of legged locomotion. In order to carry out such a plan, the logically necessary motor primitive is the ability to move the center of gravity (COO) with respect to the support base3. This can be obtained by means of two methods: (i) movements of the upper part of the body with respect to the pelvis, and (ii) movements of the pelvis with respect to the legs. The former method involves the "direct operation" of the limbs and the latter method operates the legs in the "reverse mode", i.e. it implies the ability to compute trajectories or equilibrium points not of the end part of a limb but of its origin (what we might represent with a command of the type "move pelvis".) The reverse use of the legs is functionally significant for any support phase (with one foot or two feet or even in multi - legged locomotion) and we think that it plays the leading role. The result is a zig - zag oscillation of the COO in the horizontal plane for standard level walking, similar to humen behavior. Moreover, for different footprint patterns such as stepping sideways, the COO path is different, but the control strategy remains the same. For example, In moving from one supporting foot to the other, the COO may get outside the support base resulting in a short falling phase terminated when the other foot meets its target on the ground. The footprints are also the aiming or guiding points for the feet and their co-ordination is constrained by two requirements: (a) a requirement of internal coherence, that affects mainly the timing (i.e. each next footprint can be instanced only when appropriate internal events of the walking process have occurred), (b) a requirement of spatial programming, which means to adapt the foot-path to the environment and/or to the task. For example, the presence of an obstacle, which can be overcome without stepping over it, requires to adjust the standard foot-path in the neighborhood of the obstacle. In other words, the path automaton must be "cognitively penetrable" by the planning aspects of the global performance. In some cases, the foot-path may degenerate to a simplified form, as when marching in place; the representation of the walking skill must be able to manage also these cases without any ad hoc modification. The balance automaton is intended to deal with a variety of postural adjustments that improve the dynamic balance of the walking pattern, for example the swinging movements of the arms, which counteract the dynamic disturbances due to the swinging legs, and the rotation of the shoulder girdle around a vertical axis, which counteracts the effects due to the rotation of the pelvic girdle in the opposite direction. The latter movement has also the functional purpose of increasing the step length and decreasing the vertical oscillation of the COG. Other subtler active postural adjustments occur in the walking pattern and they could be added to the performance of 3 The support base is the convex planar surface which contains the parts of the body contacting the ground. The condition 01 stability ot a body is that a vertical line through the COG hits the ground inside the support base.

 the automaton in more accurate formalizations. In a sense, the postural adjustments summarized above may also be considered as examples of "qualitative reasoning" in Lagrangian mechanics. It is well known indeed that the dynamics of the human body is extremely complex and computationally demanding: also postural compensations have the same degree of complexity. However, it is common experience that some simple tricks provide useful even though not exact solutions. For example, the postural disturbances induced by a swinging iimb tend to be reduced if another limb swings the other way around. The two leg automata are the actual "engines" of the mechanism. They generate placing/releasing/aiming movements of the foot, plantar/dorsal flexions of the ankle and flexion/extension of the knee, in a co-ordinated and purposeful way. In particular, the peculiar trajectory of the foot during the swing may be obtained as the result of the concurrent action of (i) an "aiming" movement of the foot to its target footprint, (ii) a flexion of the leg toward the hip, and (iii) the falling due to gravity. Many different situations may occur. For example, the first action is trivial but still formally necessary when stepping on place; the second can be modulated (e.g. in overcoming an obstacle); and the third can be absent in certain situations. The result is that the same mechanism can produce an infinity of patterns. The choice of the number of automata and the detailed distribution of tasks to any of them is certainly not unique. (Perhaps, this kind of indeterminacy corresponds to the different "walking styles" which can be perceived also by the casual observer.) The following is an attempt to provide a formalization which is sufficiently representative of some walkers. As already stated, the different automata which make up the walking skill operate on the same data structure - the human skeleton. In biological terms, this can be considered as a mental/analogic representation (a dynamic body-image or body-model) which serves as a common data base for motor, cognitive, and perceptual processes. Many processes operate concurrently on the internal body model and the generation of the actual motor output (the rotations of the individual joints) can be regarded as analogous to the process of measuring or monitoring the movements of the model. From a simulation point of view (see the chapter on NEM) the situation is basically the same: The output interface, in this case, does not generate muscle contractions but graphic commands. 2.2. The Foot_Path Automaton Operation Generating guide points for the foot path Moving the pelvis in the direction ofThe foot - prints other cognitive path formation tasks

Complex Motor Patterns: Walking 67 States RP: generation of a guide point for the right foot; LP: generation of a guide point for the left foot; RE: rest; Transition Diagram < - LP > Rp < > RE < > Semantics LP: 'change" to RE II a "stop walking" signal is given, else rcompute" a guide point lor the left loot: "move* the pelvis toward It; "change" to RP when the "lelt_loot_on" signal is given). RP: "change" to RE II a "stop walking" signal is given, else I'compute" a guide point lor the right loot: "move" the pelvis toward it; "change" to LP when the "right_loot_on" signal is given). RE: "compute" the last step; "wait" until a "start walking" command is given; il the command is to the right then "change" state to RP, otherwise "change" to LP. The 'compute' operation is the part that must be cognitively penetrable by many different conditions: the nature of the task and geometric relations such as the position of the foot with respect to targets, obstacles, etc. This means that the "compute' operation must be programmable and cannot be wired-in at spinal level: its programming is the responsibility of a 'planner' with which the automaton must share a common language and an effective interface. In particular, the specific computation needed when the "stop walking* signal Is given consists of selecting a guide point for the foot which is behind to meet the foot which Is in front. It is worth noting that the automaton is not an "oscillator" but it will produce oscillatory patterns in the case of regular patterns of planning, such as In level walking. In other cases, the periodicity may disappear, even if the basic alternation still remains.

68 L Baratto, P. Morasso and R. Zaccaria The "move" command to the pelvis implies a reverse operation of the legs. Note that the command is a "move" and not an "aim". The former denotes a relative motion along a specified direction, whereas the latter denotes an absolute motion toward a specified target. The relevant difference concerns composition with other commands. For example, the move command to the pelvis, together with the effect of the dorsal flexion of the ankle In the pushing phase of the step, generates the characteristic oscillation of the COG on the vertical plane for standard straight walking. The automaton is not self-paced: It needs the feedback about the "foot-strike" event. Note that such event will be different for different walking Instances. When walking forward it will be "heel-strike", when walking backward it will be "toe-strike", when walking sideways it will be "side-strike". In general, a formalization of skills must be generic enough to manage differences of this sort. The synchronization mechanism is also an example of a rather general principle of motor organization. A fragment of motor program naturally produces the expectation of events which focus the perceptual attention on specific sensory patterns. During walking, the support base moves and changes its shape (In level walking It follows a zig-zag pattern.) During single support it coincides with the supporting foot. After the swinging foot strikes the ground it extends and tilts toward it and so on. The task of the automaton is to drive the COG over It or, at least, to assure that after small falling phases the COG reenters in the extended base. In biped locomotion the task is critical because the base is narrow; in mul- tilegged locomotion the task, although easier, is formally the same. 2.3. The Balance Automaton Operations . rotating the pelvic girdle . rotating the shoulder girdle . swinging the arms . other balancing actions States RS: the right arm swings forward; LS: the left arm swings forward; RE: rest.

Complex Motor Patterns: Walking 69 State Transition Diagram RS > LS > RE < Semantics RS: "chenge" to RE if the "stop walking" signal is givan. else I'start swinging" the right arm forward and the left arm backward & "start turning' the pelvic girdle counterclockwise & "start turning* the shoulder girdle clockwise & "chenge" to LS" when the "foot - strike" signal is given] when the signal <the left loot is off and swings forward> or the signal <the

right loot is off and swings backward> is givan. LS: "change" to RE il the 'stop walking" signal is given, else I'start swinging" the left arm lorward and the right arm backward & "start turning" the pelvic girdle clockwise & "8tert turning" the shoulder girdle counterclockwise & "chenge" to RS* when the "loot-strike" signal is given] when the signal <the right loot is off and swings forward > or the signal <the left foot is off and swings backward >. RE: "wait" until a "start-walking' signal is given: if the direction is forward and the initial loot is left then "Changs" to RS, otherwise change to LS. '&' stands lor parallel execution, ';' stands lor sequentlel execution. During forward motion, pelvic rotation displaces forward the hip of the swinging leg and backward the hip of the supporting leg. It affects the movements of the two legs, reducing the angular excursions or, for the same angular excursion, increasing the length of the step4. The balancing pattern depends on the direction of locomotion. For example, when moving sideways the swinging and turning movements are inappropriate. The signal which triggers them must be selective enough to understand this situation and to filter it out. The actual direction of movement with respect to the body and its changes in complex locomotion patterns need not be explicitly stated because they are implicitly contained in the incremental generation of the foot-path. The suppression of the dynamic balencing patterns of the arms, shoulders, and pelvis leads to the inability to progress in a straight line at higher speeds of walking and to increased energy * The pelvic rotation around a vertical axis is usually associated with 'palvic list" - a slight drop of the pelvis on the non- weight -bearing side. We did not put it in the model because, although functionally significant (it contributes to reduce the vertical oscillations ol the COG) it is, perhaps, a passive effect. Passive effects are also likely to contribute to arm swing, pelvic rotation, and shoulder rotation. These movements, however, are much more "cognitively penetrable".

70 L Baratto, P. Morasso and R. Zaccaria expenditure through greater muscular effort. 2.4. The Left/Right Leg Automata Operation aiming <the swinging leg> flexing <the swinging leg> loading <the supporting leg> unloading <the supporting leg> other step formation actions. SU: the leg "supports" the body weight; SW: the leg "swings" freely; RE: rest State Transition Diagram < SU > SW > RE > Semantics5 SW: "aim" "this loot" at "this loot-print" & "flex" "this leg" & "change* to SU when "this loot" strikes the ground. SU: "place- "this loot" on the ground; "change" to RE II the "stop walking" signal is given, else |"start releasing' "this foot" when the COG goes outside 'this foot" & "change"

to SW when "that loot" has been placed to the ground).

Complex Motor Patterns: Walking 71 RE: "change" to SW when the "start walking this loot" command Is given. The swinging movement of the leg begins with when the other foot is placed on the ground. Then by the concurrent activation of two processes the movement starts. One process aims the foot at its target location and the other "shortens" the leg (co-ordinated hip/knee/ankle flexion) in order to avoid shuffling. The latter movement can aleo be tuned to overcome small obstacles without changing the pattern. The length of the step covered during swinging is known to be linearly related to the step rate6. The duration of the swinging movement is dependent on factors which are under control such as the speed of the aiming movement, and factors which are not such as the forward falling phase and irregularities in the terrain. This duration is constrained by the need of co-ordination with the programmed movement of the pelvis7. During the swing phase of walking, the thigh, shank, and foot are free in space and can rotate in the internal direction, working with the rotation of the pelvic girdle. The stance or support phase is more complex because there is interaction with the ground. At ieast three steges can be distinguished: (i) the initial stage of placing the foot on the ground and gradually loading it, (ii) the stage in which the leg is fully loaded and carries out its locomotor function operating in the reverse mode, and (ili) the final stage in which the contact with the ground is gradually released and the foot is made ready for takeoff. Loading and unloading go together with placing and releasing because conourrently the COO is driven from one foot-print to another, in anticipation. In the placing phase, the foot is driven In such a way as to loose progressively all Its degrees of freedom. In moving forward, the heel is the first to be fixed and the toe Is the last, whereas in moving backward it is the other way around. "Fixing" has both a motor and a perceptual meaning. It means to operate the local degrees of freedom to avoid mutual movements between a part of the body and the ground and to monitor that slippage does not occur. Humans heavily depend on the presence of friction for carrying out foot placement. This means that the computations are not exact but only approximated and perhaps depend on such analog mechanisms as the compliance of the muscles and segmental reflexes which give appropriate responses if the anticipatory movements of the COG guarantee the right loading and unloading. At the end of the placing phase, which can be decoded by some sensorial computation, the COG should have entered the loaded foot and should remain inside it until the beginning of the 5 The two leg automata are obviously symmetric. We write the lormel structure ot only ona ot them. Instead ot "iafl" and "right", we use "this" and "thet" to denota the two feet or legs. "This" corresponds to the same automaton and "that" to the other automaton. It Is necessary to use such a convention because the two automata interact. (One is ayn- chronizad on events generated by the other.) 6 The stride langth, i.e. the length ot the lull walking cycle ot two staps, depends on both the step Irequency and the leg length (or also the body height). Oean [1966] Iran a large body ol data found that lor male subjects the loflowlng relation gave a good fit: (stride length/body heightystep rate - 0.008 lor step rates In the range 80-130 staps par minute. 7 Since the movement of the pelvis anticipates the swinging leg, the programmed speed of the lormer should ba somewhat smaller than the programmed speed of the latter. Moreover, the speed of swinging should be large enough to allow the swinging loot to overcome the patvis just before hitting the ground. The value of the ratio is not critical, anyway, because the two automata are cross synchronized: the loot_path automaton doss not start moving the parvls to a new loot-print until the swinging leg has hit the ground and the swinging leg is aimed at the current loot-print which is attracting also the pelvis.

 final phase. In the middle part of stance, maintaining the foot stationary on the floor without any slippage requires that the shank can rotate with respect to the foot In all the planes. (One angular component is mainly determined by the translation of the pelvis and the other two by the requirement to compensate for the rotations of the pelvic girdle8.) However, such a complex co - ordination of rotation patterns does not need to be programmed explicitly: It is Implicitly contained in the reverse operation of the leg kinematic chain. The final phase is characterized by a progressive release of the degrees of freedom of the foot In an order opposite to the placing phase. It is started by some sensorial decoding of the event which marks the exit of the COO from the foot. In forward walking in particular, this is implemented with a plantar flexion of the ankle which pushes up the COG and contributes to the characteristic vertical oscillation. This phase is ended when the other foot has terminated its placing. During the first part of this phase, the COQ may get outside the support base, initiating a small falling movement which is terminated when the other foot touches down and starts its placing. The support base changes constantly its shape, area, and orientation and the whole walking pattern may be characterized as "the base chasing the COG". It is also worth noting that foot placing and loading, releasing and unloading have much in common with many manipulation skills which involve interaction with the environment such as grasping and brush painting. In both cases, it is the uncertainty about the body/environment interface which makes exact control mechanisms inappropriate and forces the brain to choose approximate solutions. 2.5. The Overall Skill The walking skill is the concurrent operation over the body model of the four internal automata and of a fifth external automaton - the environment. Foot-Path Balance Automaton Automaton Left Leg Body Right Leg Automaton Model Automaton Environment The automata communicate by means of messages like "right heel down" and share global Information such as the guiding foot-prints. The environment sends and receives "messages" too, such as the pull of gravity, the forces of contact, and the forces due to friction. In other words, the environment may be considered as just another "actor" (or group of actors) in the 8 The actual rotations between shank and loot are made possible, In addition to the ankle Joint, also by articulations ot the loot, e.g. the subtalar joint. On the other hand, we ihink that such a comptax mechanical arrangement is not oruclal Ifom the cognitive/representational point ot view.

Complex Motor Patterns: Walking 73 "choreography" of the walking skill. In the case of a computer simulation or computer animation, as in NEM, it is natural to choose the "animistic" point of view and to implement the relevant actions/reactions of the environment as little specially programmed demons. In the special case of normal level walking, the motor performance can be characterized by a temporal walking diagram, like the one of figure 1. Once more, we remark that this is a superficial description, which does not capture the structured richness of the skill. General aspects of a skill which should not be absent in a formalization are (I) the synchronization with external events, such as music in dancing, (II) the smooth chaining with another skill such as sitting after having approached a chair, and (iii) its coherence. The synchronization problem can be seen as the interaction between the skill which Is a composite actor and a timing actor which sets deadlines and other timing constraints: Timing > Skill Actor < Actor The effect of the message passing among the two actors is to tune the timing parameters of the automata which compose the skill. The smooth chaining of a skill with another skill, without discontinuities or stopping points, may be obtained by fusing the final operation of one with the initial operation of the other. In walking, for example, the REST states of the automata are naturally suited for that. heel foot heel toe heel contact flat ri»« off contact 20 30 40 50 60 70 80 90 S88SMMSB8BSBS SBBS8SS88SS8SSSSSSSSSSSSSSSS6SSSSS5SSSSSSBSBSSSSSSSSSS 20 30 40 SO 60 70 00 90 100 heal toe heel foot heel off contact flat ri" Figure 1. Temporal Walking Diagram

 3. DESCRIBING AND MODELING CENTRAL DISORDERS OF MOVEMENT The formal description of complex movement patterns has a long history in the neurologic clinic because its alterations are signs or symptoms of neural lesions [Jackson 1884, Brain and Walton 1969, Tardieau 1973, Knuttson and Richards 1979, Grimm 1983]. In particular, one of the patterns which has received the greatest attention is walking, possibly because this is a pattern that involves the participation of many neural subsystems and is very sensitive to their cooperation and integration [Qrillner 1975, Conrad et al. 1983]. Therefore, we think that a discussion of the pathological alterations of gait, contingent to lesions of the central nervous system (CNS), has a very significant epistemological value9. We think indeed that, in addition to an improvement of the instrumental techniques of analysis, the whole field of movement pathology would greatly benefit from an interaction with the current theoretical and computational investigations from Cognitive Science and Artificial Intelligence. 3.1. Pathological Gait Syndromes in Clinical Neurology The oldest and most significant descriptions of pathological gait patterns in lesions of the CNS can be found in manuals of neurology. Indeed, the usa of signs coming from the direct observation of the motor performance of the patients was the only diagnostic tool available to identify the nature and the location of a neural disease, in a time when the biochemistry of the CNS was unknown and neuroradiologic techniques were not available. In particular, the observed modifications of gait, which proved to be consistent signs of central neurolesions, have been organized and classified In different syndromes that we briefly describe in the following. A general distinction must be made between one-sided disturbances or hemisyndromes and symmetric disturbances or parasyndromes, which affect both legs. In hemisyndromes, also the unaffected limb may exhibit pseudo - pathological kinematics due to compensation mechanisms and the disabled limb may show a disturbed pattern different from that seen in bilateral disabilities of the same nature. Hemlptegic Gait Syndrome In the case of hemiplegia, which is classically attributed to

lesions of long descending pathways through the pyramids of the medulla, the characteristic aspect of the gait pattern Is described as "mowing". This pathological kinematic modification is caused by the extended contraction of the paretic leg because it makes Impossible, during the swing phase, the shortening of ankle-hip distance normally produced by the concurrent flexions of the hip and knee joints. As a consequence, the hemiplegic patient In walking circumducts his paralysed leg (with the knee fully * As suggested by Grimm [1983], movement disorders may ba classified as program or central disorders (determined by alterations above the level ol motor units and beyond the central terminations of primary alferents) and non - program or peripheral disorders where the CNS is normal, with deficit existing in the motor apparatus.

Complex Motor Patterns: Walking 75 extended and the ankle plantar-flexed: equinu£) swinging It outward at the hip, like a sickle. While performing the "mowing movement", the paretic foot, which is kept in full plantar flexion, grazes the ground with the tip and the side. In extreme cases of the equinus, the gait pattern is disrupted in its basic symmetry: The step size is greatly reduced and the normal foot lags behind the paretic one, never passing it. The duration of the stance phase of the paretic leg is always shorter than that of the normal leg. It is worth noting that the latter kind of asymmetry may not be present in pathologies in which the same missing activation of the flexor muscles of the leg does not depend on a central lesion but is the result of a peripheral deficit. This is the case, for example, of poliomyelitis which destroys the motor neurons in the spinal cord without affecting the CNS. Even marked differences of muscular strength between the normal limb and the paretic one do not alter the symmetry of the duration of the stance phases. Therefore, the asymmetry of the hemiplegic gait is likely to be a true "program disorder". Parkinsonian Gait Syndrome Parkinsonism is a disease which is determined by a neurochemical pathology of the nigrostriatal striatonigral system (the mutual interactions of the substantia nigra and of the striatum). In a Parkinsonian patient who has a bilateral deficit, the initiation of walking is uncertain and slow. The patient appears like "nailed to the floor". He is hesitant and only after many attempts he succeeds to trigger the whole walking pattern. A similar aklneslc feature is displayed when the subject is required to change direction. The rhythm is usually slow (bradykinesia) and the steps are small and shuffling. However, either spontaneously or as a response to being pushed forward, the

Parkinsonian patient may exhibit a progressive increase of the pace, hurrying his steps and leaning his trunk forward as if "trying to catch up to his center of gravity". When this happens, stopping becomes difficult and uncertain, mirroring the difficulty in the starting phase. Another striking phenomenon is the so called kinesia paradoxical. In particular situations of stress, as in the case of an immediate danger, the Parkinsonian patient may be able to switch from the uncertain and slow gait to fast walking or even running. Other motor disorders in Parkinsonian patients are (i) rigidity, which is enhanced by arousal, (li) tremor, which is suppressed by voluntary actions, (iii) inability to modulate the myotactic reflex gain, (iv) loss of fusimotor muscle spindle control, (v) motor unit recruitment pattern errors, and (vi) inability to modulate EMQ burst durations. On the other hand, these patients show substantially normal postural responses. Cerebellar Gait Syndrome A patient with lesions of the cerebellum walks like a "drunk man", i.e. like one who has a severe balancing disorder and, therefore, fear of falling10. They keep their legs spread apart, to increase 10 The neurophysiologies! bases ol this disorder were investigated by Nashner and Qrim |197B) In a study of the postural responses ot cerebellar subtects on a platlorm. They lound (i) absence ot the myotactic stretch rellax. (II) abnormality in the structure and the metric* ot the long latency stretch responses, (iii) a long delay In the occurrence ot compensatory vestibular responses, although the structure ol the response, i.e. the selection ol the muscle group, was preserved.

 the base of support, and the arms are abducted, to improve stability11. The affected lower limbs of cerebellar patients are markedly ataxic. Muscular contractions are weak, more easily fatigued than normally, and they are characterized by irregularity, intermittency, and delays both In initiating and in terminating contractions. The gait pattern lacks smoothness: The feet are kicked forward ballistically, with uneven rhythm and size. As a consequence, the pathway is irregular, a sort of zig-zag pattern. The patient with a unilateral lesion tends to stagger toward the affected side and to deviate toward this side in walking. This may be well demonstrated by asking the patient to walk around a chair. When the affected part is inside the path, he tends to fall into the chair, whereas in the opposite case he tends to move away from the chair in a spiral, The Motoscopic Exam The described alterations of complex motor patterns like those summarized above find clinical application in the so called motoscopic exam, which still has practical significance because even small behavioral signs can provide an early indication of a specific pathology. However, the diagnostic trend In modern clinical neurology Is to rely more and more on instrumental techniques, reducing the need to improve the power and the precision of motoscopy. This is due to the fact that the diagnostic problem is concerned directly with the localization and possibly the nature of a neural lesion and only indirectly with the functional motor deficit. Therefore, in the diagnostic context (which Is presently the main stream of clinical neurology) there is very limited interest to further develop the observation of normal and abnormal movement patterns in such a way to understand the underlying organization of the neural motor control system. We cannot forget, of course, the large body of research in the field of neurophysiology in general and of neurophysiopathology in particular, centered on the basic mechanisms of motor control. The problem here is the level of analysis, which is extremely "microscopic" and far remote from the need of understanding the nature of functional motor deficits in everyday activities. On the contrary, the crucial need of improving the objective observation of common movement patterns and of interpreting them in the context of a model of the motor control system is clearly seen in the field of motor rehabilitation. In this case, indeed, the motor recovery and functional re-education are based on a highly pragmatic approach which has the only choice of "educating or re-educating to movement by means of movement". Movement is both the goal and the tool of therapy. 11 The improvement of stability due to the abducted position of the arms, or to the use ol balance beams as in the case ol tight rope walkers, is releted to the moment ol inertia ol the body. Abducting the arms increases the moment ol inertia ot the body and therelore increases its "resistance" to disturbances.

Complex Motor Patterns: Walking 77 3.2. Rehabilitation Clinic: Observation of Pathological Movements The clinical observation of movement pathology, in the context of the rehabilitation experience, although initially inspired by the classic neurological clinic, has promptly singled out its descriptive limits. Again, gait is a useful paradigm which well represents the range of motor activities of interest and which has been the object of a large number of critical observations. Brunnstrom [1970] was one of the first to point out that the "mowing gait" is a frequent but not universal pattern in hemiplegic walking. An alternative pattern, that may substitute the previous one particularly in the swing phase of gait, is characterized by a concurrent flexion at the hip, knee, and ankle. During the same years, Bobath and Bobath [1975] were carefully observing the walking deficiencies in children who were affected by lesions of the CNS in the pari - natal period. These observations demonstrated that the static concept of pathological motor patterns, formalized in the classic gait syndrome, was unacceptable. On the contrary, the pathological patterns are adaptive and are transformed, during development and/or recovery: the effect of uncontrolled activations of muscle groups is countered by means of compensatory patterns that improve the functional performance like in the speed and stability of walking. Moreover, Bobaths were the first to suggest, from the evidence of observed behavior, that in pathological movements it is possible to find the presence of primeval patterns, that is early patterns in the ontogenis of the motor system. These patterns, although integral parts of the pathological behavior, are not expressions of clear neuromotor deficits, like spasticity or tremor, and would be quite normal at the appropriate level of development. Their disturbing effect, which contributes to the functional deficit, is a strong stereotipicity and also a lack of co-ordination with other concurrent motor patterns which may be related to other stages of development. For example, a typical primeval behavior is the so called "asymmetric tonic neck reflex". This is characterized by the concurrent extension of the limbs on one side of the body and the flexion of the limbs on the other side. The reflex is thought to be triggered by the neck receptors during the rotation of the head in the direction of the extended arm. This behavior is indeed normal in babies until the 16th week of age. If it persists indefinitely, it becomes a noxious component of pathological movements. However, such primeval schemata can be exploited for a functional purpose by subjects with a central neurolesion, representing so the adaptation of the impaired motor control system. For example, children affected by involuntary contractions in flexion of the legs during standing or walking may voluntarily use the asymmetric tonic neck reflex to contrast this tendency. With regard to adult patients, it is interesting to summarize a contribution by Lissoni and Boccardi [1982] who, developing previous observations by Knuttson [1979] on the basis of motos- copic data, proposed an articulated interpretation of the knee hyperextension in hemiplegic walking (genu recurvatum). In contrast with classic neurology, they think that this phenomenon corresponds to four quite different situations which can be discriminated by looking at the time of initiation of the knee hyperextension. If the extension precedes the heel strike, it is considered a programmed (pathological) extension of the whole leg in line with the classical neurologic theory. If the extension is increased immediately after heel strike, one possible interpretation is that the

 genu recurvatum is in fact a postural adjustment chosen by the CNS In order to compensate for a lack of control of the contraction of the plegic quadriceps muscle. As a third possibility, if the hyperextension is triggered in the initial part of stance, the pathological cause is probably a spastic response of the triceps surae[12], stretched by the forward motion of the knee with respect to the foot. The shortening of the muscle is supposed to be violent enough to stop the knee, which is then hyperextended by the concurrent forward motion of the hip. Finally, a hyperextension which occurs in the middle part of stance may be considered as the activation of a primeval reaction, called "positive support reaction", which generates an extension/flexion pattern similar to the tonic asymmetric neck reflex, contingent to a tactile stimulation of the foot plant.

3.3. Rehabilitation Clinic: the Motor Interactive Model

In general, in addition to the effort for a better descriptive analysis, a trend in rehabilitation is to formulate models of pathological behaviors not too strictly dependent on the pure anatomical location of the neural lesion. A key concept for the development of appropriate kinesiologlcal models is that the observed patterns of movement depend on the interaction between "internal" patterns of muscle activation and "external" forces and constraints, which include both the biomechanical and the environmental constraints. In the end, the result Is an equilibrium between Internal and external forces but it is also clear that many subtle mechanisms contribute to it. In particular, the "internal forces" do not depend only on the patterns of muscular activity but also on the actual movement because muscles are "compliant actuators* (see the chapter on compliance), and the patterns of muscular activity are functions of the central programs as well as of the sensory feedback, which in turns measures the actual movement again. In other words, if we look at movement as the result of an equilibrium of forces we realize that movement is an implicit function of movement itself, in which neural, neuromuscular, and biomechanical aspects are coupled In a very tight way[13]. In short, we may speak of an interactive model of movement production. In simple cases, the equilibrium configuration may be reached passively. This is the case, for example, of the progressive fixation of the ankle in plantar flexion in patients forced to bed for a long time by an hemiplegia. During the first few weeks after the ictus, indeed, the flaccid paralysis of the tibialis anterior does not obstacle the pull of gravity to a full plantar flexion of the foot, further increased by the progressive appearance of a spastic tension of the triceps. If this position is maintained too long, it induces a modification of the muscle lengths and of the capsular/ligamental structures, which "fixate passively" the new pathological configuration. In many cases, however, a pathological pattern is the result of the "adaptive search" of an appropriate equilibrium: it may happen indeed that the patient "attempts" to solve static and/or dynamic problems which are functionally important by exploiting pathological patterns. 12 The lunction ol the triceps surae is plantar llexion ol the loot. 13 This is a concept that was clearly identilied by Bernstein |1957|.

Complex Motor Patterns: Walking 79 The "interactive" modeling of normal and pathological movements is still a work in progress which needs further experimentation arid critical analysis. For example, Tardieau pointed out already [Tardieau 1973] that the same "movement pattern" might be compoeed of different patterns of muscle activities in different subjects. In the same line of reasoning, we may quote the results of experiments of gait analysis by Boccardi and Pedotti [1982]. They showed that in some hemiplegic subjects the apparent quasi - normality of the gait patterns during walking was obtained in the final part of stance by the "clever" co-ordination, of a pathological phenomenon - the clonus of the triceps surae muscle - and of the outcome of a normal phenomenon - the increased elastic force of the same muscle due to a slight reduction of its length. Even if the muscle was paralyzed, the combination of the two phenomena was functionally effective, allowing the heel to take off the ground while resisting gravity. From this observations, one might conclude that in studying motor control impairments the presence of certain pathological patterns or even their absence in specific experimental conditions do not have an absolute meaning. On the contrary, what is essential is the capability of the CNS to solve a "generic" motor task in a variety of environmental conditions. It is clear indeed that, in the experiments by Boccardi and Pedotti referenced above, small modifications of the paradigms would heve inevitably singled out the presence of a paralyzed triceps, i.e. the inability of the CNS to implement the locomotion function in variable conditions. We think that future improvements of the motoecopic observation of pathologic motor behavior cannot Ignore this central point. 3.4. Rehabilitation Clinic: the Motor Cognitive Model The adaptive performance of subjects who attempt to recover motor functions after lesions of the CNS may be the object of two possible interpretations: a "local" interpretation and a "general" interpretation . The first one is that different lesions determine different types of "error". Consequently, adaptive adjustments may vary widely as a function of the specific "error" and in particular situations the adjustments may be so effective to reproduce the normal kinematics. The second interpretation is related to the common experience of rehabilitators that patients with central neural lesions have a clear tendency to stick to a few stereotyped patterns and once organized they become so overwhelming to inhibit also residual normal capacities. One example, in the case of early central lesions like cerebral palsy, is the persistence of primeval motor patterns in addition to clear pathological motor signs such as spasticity, tremor, and rigidity. These data are compatible with the hypothesis that, in addition to "specific" motor symptoms, a significant consequence of a central motor lesion is the difficulty to organize "general" motor plans which can be applied to a variety of specific but similar problems. In turn, a general motor plan requires an internal or mental model parametrized in such a way to be adaptable to variable circumstances. A model of locomotion such as that discussed in the first part of the chapter is an example; in an early stage we could think of a more primitive model which manages only the postural adjustments that guarantee standing, etc.

The validity of the Internal model hypothesis is also supported by the clinic observations of Milani Comparetti and Qidoni (1971) about primeval patterns: Their persistence, according to the authors, inhibits the development of newer and better motor patterns. For example, the "prehension reflex"14 must disappear in order to develop the ability to lift a trunk over the forearms. Similarly, the "asymmetric tonic neck reflex" must be overcome by the "trunk derotative straightening reaction"15 and the "symmetric tonic neck reflex"16 must come under control before the quadruped locomotion may be attempted. Milani Comparetti thinks that the postural maturation of the CNS relies on the fact that, at some stage of development, some reactions become "overwhelming", that is, they become more easily triggered than others: They play the role of "organizers" of the overall motor skill. Later in the development, when better organizing schemata become possible, the "trigger level" of these reactions is depressed, and they remain latent but available. So, primeval patterns which have the role of breaking the original fetal patterns must become less overwhelming in order to be substituted by a steady sequence of "organizing patterns" which drive the motor maturation from sitting for standing and walking. If we consider the tendency of subjects with central neuro - lesions to reduce the number and the degree of flexibility of their motor patterns, the simplest explanation for this tendency could be a smaller or larger extension of the damage to the executive motor mechanisms. This is a constraint which reduces the range of choices. However, we think that this is too simplistic. The observation of the pathological behavior of patients with cerebral lesions reveals quite clearly a difficulty in initiating the acquisition of new behaviors, a phase of learning that has been named "exploratory" by Le Boulch [1971] in his study of the learning techniques of complex gestures. The normal subject, in this phase, has a redundant behavior, with imprecise, useless or even contradictory movements. On the other hand, it is this initial phase which allows the subject to face the global complexity of the task, in concrete terms: The systematic experimentation is the necessary condition to identify and abstract the characteristic and common features of the motor structure and to create a flexible motor schema, fit to variations and generalizations17. When this phase is completed, then it is possible to proceed to a phase of "refinement" in which the movement schema (Le Boulch calls it "global movement shape") is "edited", dissociating ineffective automatisms, tuning the parameters of some movements etc. on the basis of a scale of performance. Patients with central lesions have clear difficulties in learning whereas this is not the case for patients whose pathological movements are due to insufficient muscular force, to limited joint excursion or to other peripheral impairments. With these patients it is not uncommon to find the capability to learn remarkable motor tasks, expressing a high degree of "motor intelligence". 14 One of the aspects ol the "prehension rellex" is the llexion of the lingers, In a pronated hand, triggered by the contact ot the palm with a support surface. 15 The "trunk derotettve straightening reaction" consists ol a tendency to restore the alignment of the vertebral column as a rasponse ol the rotation ol one ol its segments (head, shoulder girdle, pelvic girdia). 16 The "symmetric tonic neck rellex" is a reaction ol the limbs to the stimuletlon ot the neck receptors: the backward extension ol the head triggers the extension ol the upper limbs, whereas the lorward llexion causes the opposite pattern. 17 Although this procedure involves trial and error, it should not be contused with the "trlel and error learning modality" as it is conceived in experimental psychology. In this modality, indeed, the representation ol a movement sequence which solves a problem, i.e. a skill, follows the learning phase and is scarcely fit to generalization. On the contrary, in the learning paradigm described by Le Boulch the abstract representation ol the problem precedes the experimental activity which leads to learning.

 In our opinion, these fundamental characteristics of the motor system can be attributed to those cognitive processes which are capable of expressing spatio-temporal patterns appropriate to interactions between subjects and their environment. It is also possible to formulate the hypothesis that central lesions decrease the "efficiency" of this "cognitive function" and we could re - interpret the classical descriptions of neuromotor pathology as expressions of a disorganization of motor subsystems, secondary to the mentioned decrease of the cognitive function. Already a traditional concept is the interpretation of the pathological signs of movements in central lesions of the CNS as phenomena of "liberation" of lower centers from the control of impaired higher centers. Also partially as phenomena of "re-organization", these centers act according to simpler operation models. For example, one of the first formalizations of this concept was done by Jackson [1884] who theorized three levels of representation of the muscles in the CNS, in order of increasing complexity: The lowest one would allow only elementary combinations for the most automatic movements and the highest one would be related to the flexible combinations typical of voluntary movements. In particular, according to Jackson, hemiplegia is characterized by a damage to the intermediate level with a consequent liberation and rearrangement of the low level which would justify, for example, the appearance of "synkinetic movements"18. Although the formalization of Jackson has been widely criticized, the broad interpretation of the effect of central lesions on the neural "hardware" is still accepted, particularly in the neurological clinic. Our attention, on the contrary, was focused on the formalization of movement as a cognitive process and on its use for evaluating objective descriptions of pathological movement patterns. Moreover, the close association of theories of motor control with cognitive theories has been emphasized recently by Stelmach and Hughes [1984]. We think that the general implication is to shift the focus of attention in "movement understanding" from muscle patterns to mental action models. 18 In a hemiplegic patient, synkinetic movements may ba triggered either passively, due to an elective stimulus, or voluntarily. For example, a llexion 01 the lorearm is inveriably associated with Its pronation, the ulnar liaxion of the wrist, and the flexion ol the lingers to a fist.

This Page Intentionally Left Blank

PARTI Chapter 3 DANCE AND MOVEMENT NOTATION Antonio Camurri, Pietro Morasso, Vincenzo Tagliasco, and Renato Zaccaria CONTENTS

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasco (Editors) © Elsevier Science Publishers B.V. (North-Holland), 1986 85 DANCE AND MOVEMENT NOTATION Antonio Camurri, Pietro Morasso, Vincenzo Tagliasco, and Renato Zaccaria Dance is a complex motor activity that strongly emphasizes the holistic nature of movement. Dance notation is an attempt to express this in symbols. However this is difficult because many different types of representation may be needed in different circumstances. Many different notation systems, for dance as well as for other formalized types of movements, were indeed invented over the centuries but they miss the need to consider motor patterns from different points of view and with different levels of detsil. A computational approach to dance notation is discussed, which is centered on the multiplicity of representations and on the transformations among them. 1. A COMPUTATIONAL APPROACH TO DANCE NOTATION Dance is a complex motor activity which strongly emphasizes the holistic nature of movement. The fact is that the interplay among the motor processes is by far more crucial than the particular operation of any and each of them. The result is an apparent global simplicity that implies an extremely complex network of interactions - interactions among different motor subsystems, among motor and perceptual subsystems, among the actor and the environment, etc. The "superficial" global simplicity of movement may be related to the cognitive/planning aspects of movement, which emphasize the role of the whole body as a tool to fulfill a task. For example, a task such as "pick up something" implies, for a planner, an interest in the motion of the hand and not in the co-ordination among the arm, the torso, and the legs. From this point of view, the extreme complexity underlying any motor act is the necessary prerequisite to carry out a task without constraining it. In other words, we deem it useful to turn around the common approach to motor complexity that focuses the attention on the role of synergies as tools used by the Central Nervous System (CNS) for decreasing the complexity of motor control. On the contrary, a sufficiently powerful "machine" is what the CNS needs in order to carry out a generic motor task.

86 A Camurri et al. Dance/Movement Notation is an attempt to express in symbols the "holistic" nature of movement, while trying to tackle the observation of movement In a scientific, unambiguous way. However, this is far from straightforward, not only for objective reasons (movement complexity) but also for subjective reasons, because the efficacy of the notation also depends on the user or the community of users. Among the possible "users" are the the audience, the choreographer, the dancer, the notator, the coach, the judge of a sportive competition. Of course, their role is quite different: (a) The audience can well be unaware of dance notation, whereas the notation should take the audience into account In an explicit way1; (b) the choreographer can use the notation to compose an action pattern In a symbolic way, i.e. without the actual dancers; (c) the dancer can use the notation to learn or rehearse an action; (d) the notator can use the notation to express in a symbolic way what his/her eyes have seen of an actual performance (notation « coding and storing language); (e) the coach can use the notation to spot the key passages of a routine and to focus attention on their timing and on the expected attributes of each of them; (f) the judge of a competion can use the notation in a similar way, with the complementary purpose of quantifying the actual performance. The emphasis of the different user classes is likely to be put on quite different aspects: (a) The audience is likely to be particularly concerned with global visual aspects of the performance, such as particular visual effects strongly dependent on the point of view, the apparent shape of paths followed by each dancer and by the patterns formed by groups of dancers (the so called floor plans as seen by the audience); (b) the choreographer needs to express, experiment , and communicate a more abstract action concept which evolves in space and time. The visual aspects at the audience level, even if important, are just a

consequence of the spatlo- temporal structure and the dancers are just devices to implement spatio-temporal ideas; (c) a dancer needs a very detailed representation of the required movement pattern, strongly oriented to the "hardware" that the CNS is supposed to control, i.e. joints and muscles, and is relatively uninterested with what the audience sees and what the choreographer thinks; (d) a notator is a very particular element of the audience, i.e. he/she is an observer who must mimic explicitly what the human visual system does implicitly, to transform time-varying imagery (what meets the eyes) Into a spatial-symbolic description. On passing, it is important to point out that what is said for dance is true in general for any type of motor activity in everyday life and in sport. The difference is that In dance there is a greater motivation to spell It out. ' The audience is an ensemble ol viewers and each viewer looks at the same performance In a different way. The choreographer or the notator must take Into account where the 'average* user Is.

Dance and Movement Notation 87 There are of course relations among the different users of dance and movement notation, but nonetheless it is clear that a dance pattern can be many different things at the same time and therefore a dance notation system should be able to swltoh from one type of reprseentation to another with sufficient ease. Such an ability to deal with multiple schemes of representation of the same phenomena is likely to be embodied in the "talent" and the "experience" of great choreograpers and dancers. However, relying on this kind of memory only has obvious drawbacks, because it leaves out an enormous amount of knowledge that remains unavailable for the community of users; that is enough motivation to try to devise methods which can capture the essential aspects of dance In a symbolic, explicit way. In fact, tens of dance notation systems have been invented during the centuries, particularly after the Renaissance. Practical reasons, naturally enough, have forced the Inventors to choose a particular organizing principle, or a particular viewpoint, or a particular emphasis for their notation, with the result of shadowing other aspects. For example, the Feuillet notation (Figure 1 A) is organized along the path on the floor to be followed by the dancer. The Stepanov system stresses the points of similarity between music and dance by using "movement notes." The Laban system (Figure 1 B) Is particularly concerned with the movement in space of the limbs. The Benesh system expresses the paths in space of the body extremities, and the Eshkol - Wachman notation expresses in a detailed way the rotations of the joints. Some systems look at the dancers from the audience point of view, some with a bird's eye and some from a dancer's point of view. A completely different approach can be found in the Eastern cultural heritage as we can see in a wide range of formalized motor activities, from yoga to the martial arts. In the next section two examples of the Eastern and Western approaches are examined in greater detail: T'ai Chi Ch'uan and the Bshkol - Wachman notation. The latter is a recent and particularly representative prototype of the western "analytic" attitude whereas the former is a very good example of the "holistic" nature of the eastern culture and, quite remarkably considering its age, still enjoys a wide spread application. The bottleneck which was encountered by all the methods lies In the inadequacies of the medium. The piece of paper is obviously inadequate to host the type of multidimensionality which characterizes motor performance, differently from music which can be stored on paper quite effectively. The multidimensionality of dance is similar to the structure of a Japanese garden, which has in a limited space an infinity of landscapes, as seen by a viewer. The advent of the computer era can change the picture because the computer can provide a multidimensional representational medium, i.e. a medium in which It is possible to "transform" a representation into another and in which it is possible to vary at pleasure the "grain" of the representation. This is what we would like to call a computational approach to dance notation, i.e. an approach centered on the multiplicity of representations and on the transformations among

 them, and we would like to distinguish it from the straightforward implementation on a computer of existing notation system of the pre-computer era [Brown and Smoliar 1976, Calvert et al 1983]. The challenge for carrying out a computational dance notation project is the same which is faced in the field of knowledge representation, which is a leading area of Artificial Intelligence. The kind of knowledge that we are talking about is the knowledge about action and movement, what we may call motor knowledge. A computational dance notation system should be powerful enough to master it. In the following sections, after a more detailed presentation of two methods of movement formalization, we shall analyze a number of different aspects of motor knowledge which should be taken into account In a computational approaoh to dance/movement notation.

 Th« rauillet n*nc« Notation System Pact

Position Symbol* fnwtitrr r*/tim. frtw* **«/fri##. Tnifh* t*/#*m. JfgarUm ttfiitm. Cimfmim ttfutm. i i St*p Symbol! r r r tuff/ifi. tufnmU. tufutmti. \ \ \ The Laban Dance Notation System A-,.. Ufl Right i«fc lush, The organization of the staff i 'i t m + *T C elk... $ lo,, £ «"•' ^ .nil, £ "»«"• f '"" The body aynbols Ah Q""jji' Direction! and levels Figure 1. A (top) Example of the Feuillet Dance Notation. B (bottom) Example of the Laban Dance Notation.

 2. TWO EXAMPLES OF MOVEMENT FORMALIZATION 2.1. An Eastern Approach: T'ai Chi Ch'uan T'ai Chi Ch'uan is a formalization of movement patterns (apparently simple and physically non - demanding) which is widely practiced in China. It can be traced back to the Confucian classic texts, in particular to the "I Ching" (Ching-Classic, I-Changes, i.e. the 'Classic Text of Changes") whose origin Is located between the first and the second millennium B.C. (Baynes [1951] Is an English translation of the I Ching. Liu [1972] disousees the relation between I Ching and T'ai Chi Ch'uan, and In Porkert [1974] there is a clear introduction to the fundamentals of the Chinese way of thinking In comparison with the western cognitive paradigms.) This book presents a cosmology which involves man and nature in a system of opposing cosmic forces: "yang" vs. "yin", I.e. male vs. female, heaven vs. earth, sun vs. moon, firmness vs. yielding, head vs. tail, etc.2. In other words, the yang-yin dualism defines a basic binary representation which admits different interpretations in different contexts. From this binary element it is possible to build arbitrarily complex binary codes, as it has been popularized by modern computer science. Moreover, it is worth noting that the German mathematician and philosopher Leibniz was well aware of the I Ching and his theory of binary arithmetic, whloh is at the basis of both computer science and mathematical logic, stems from the study of that book. It Is also Interesting to note that it is possible to arrive at binary structures either following a combinatorial approach or a recursive approach. The latter, in particular, is synthetically represented by the following statement: "In yin there is again yin and yang, in yang there is again yang and yin," which finds Its remarkable graphic representation (its logo) In the famoue circle surrounding two semicircles, linked In an s - shapa. The I Ching considers three-bit codes, usually called trigrams, and six-bit codes, also known as hexagrams: 64 hexagrams are formed by considering all the possible combinations of two trigrams. A solid line ( ) represents yang, a broken line ( ) represents yin, and the lines are stacked one on the other, the bottom one being

the first. The following table shows the list of trigrams. For convenience, we use the standard binary symbols (0,1) with the conventions yang-1, yin«0, leftmost bit-top line. In the table, the binary code of each trlgram Is shown together with the roman spelling of its Chinese name, its direct meaning (as a Chinese character) and other commonly accepted equivalent meanings (or meta-meanings) in the context of the I Ching cosmology: 2 In origin, yin and yang were used to denote the shady and the aunny slopes of a mountain and then the northern and the southern banks of a river. Afterwards, the distinction was generalized to diverse complementarity aspects of natural phenomena, at first, and then of social phenomena in speculative and technical contexts. The list of yin-yang antinomies is endless. However, let us list a few more, related to technical problems: positive ve. negative, even vs. odd, conservative vs. demanding, responsive vs. aggressive, centripetal vs. centrifugal, contractive ve. expansive, structive vs. active. Porkert |1974| discusses the Implications of the yin-yang paradigms in terms of basic modes of cognition and contrasts the analytic and causa/ mode, typical of the Western culture, to the inductive and synthetic mode which characterizes the Eastern way. In particular, whiia causality is focused on timing relatlonahipa, Inductivtty emphasizes the effects Induced at a distance (a concept which has entered into Western science quite recently, with electrodynamics and the theory of llelds).

Dance and Movement Notation 91 trigram trigram direct naturel moral spatial symbol name meaning meaning meaning meaning 000 001 011 010 110 111 101 100 k'un chen tui k'an sun ch'ien II ken earth activity pleasure pit bending heaven bright to stop earth thunder oollectlon water wind sky fire mountain weakness being joyfulnees being in flexibility strength elegance firmness NE E W N SE NW S sw (Note that the order is such that, similarly to the Gray codes, only one bit changes between one code and the following one.) In an hexagram it is possible to distinguish the top and the bottom trigrams as well as the two internal trigrams (also known as nuclear trigrams) which group lines 2,3,4 and 3,4,5 respectively. The meaning of a trigram is a (multiple value) function of the yang-yin symbols and of their mutual arrangements, e.g. two yang above one yln, as In "sun", suggest the wind which flows over the earth or also flexibility, whereas one yang surrounded by two yin, as In "k'an," suggests water, lake or also being in danger. Similarly, the possible meanings of an hexagram depend on the combination and arrangement of trigrams and the same mechanism can be expanded recursively, considering sequences of hexagrams, sequences of sequences, etc. This emphasizes the fundamental characteristics of binary systems of representing complex structures by means of simple codes. Furthermore, the richness and complexity of the representation ia multiplied by the fact that each binary pattern (a line, a trigram, an hexagram, ...) can have different types of meanings; it is the context that constrains the semantic space to a coherent and possibly unique interpretation. The I Ching texts explain each one of the 64 hexagrams, line by line, trigram by trigram, discussing all the possible local and global meanings. T'al Chi Ch'uan is a much later development in the general framework provided by the I Ching and relates to the revival of Confucianism in the Sung dynasty that was brought forth particularly by the scholar Chu Hsl In the 12th century. According to Chu Hsi, the starting point for achieving the ideal state for both the individual and the community is personal spiritual cultivation. This effort Is directed toward making explicit the "T'al Chi" (literally "Supreme Ultimate Principle") that Is potentially within every human being. The approach is twofold and consists of (i) extending one's knowledge, and GO exercising attentiveness. The latter method, in particular, is at the basis of T'ai Chi Ch'uan as a carefully

 structured and principled pattern of body movements. The actual development of T'al Chi Ch'uan is probably the work of Chang Sang-feng, a magistrate still living in the early Ming dynasty who, while casually watching the combat of a crane and a snake (according to the legendary story), perceived In living form the principles of the I Chlng: the strong changing to the yielding and the yielding changing to the strong. T'ai Chi Ch'uan is said to have been developed from the codification of natural movements in a system of body exercises, within the semantic framework provided by the I Ching. In a sense, T'ai Chi Ch'uan can be considered as a motor interpretation of the I Chlng or, better, a class of possible interpretations, according to the specific motor goal (general exercise, training for some specialised motor activity such as dance or defense, etc.). For example, at the basic level the yang-yln dualism can be Interpreted as the light form or unloaded foot vs. the heavy form or the firmly placed foot. From that we obtain possible motor interpretation of the trigrams: k'un body/abdomen/being receptive ... chen foot/arousing/movement/stepplng forward ... tui being joyous/oval shaped/smashing/hurting ... k'an heart/ear/bow/wheel/snake ... sun penetrating/being gentle/high ... ch'ien strong/creative/moving forward/turning

... li empty Inside strong outslde/bird/eye/arrow ... ken legs/arms/hands/fingers/keeping still/hidden ... More complex movement descriptions, also called movement forms, can be formed by Interpreting hexagrams. It is worth emphasizing that this kind of formulation has a deep cognitive meaning: It determines a strong coupling between movement descriptions and world descriptions. In other words, any movement can be viewed as, or can be meant to be, a paraphase of some natursi phenomenon. Movement "sentences" can be formed by chaining movement hexagrams and it can be noted that in a sequence the ambiguity margin of an hexagram is reduced by the effect of context, i.e. the global constraint of movement smoothness and balance. Infinite movement sequences are possible and the different schools of T'al Chi Ch'uan teach them accordingly. For example, the canonioal form of the Yang sohool has 108 different postures, the first two of which are briefly described In the following.

Dance and Movement Notation 93 First Movement - Code 101000 The initial posture (Figure 2A) is erect, with hands at side, palms back, heels together, and toes slightly apart. The name of the hexagram is Chin, which means progress. The upper trigram is Li (sun) and the lower trigram is K'un (earth), thence the metaphorlc meaning "the sun rising above the earth." However, K'un means also "abdomen," Li may depict a qualitative feature of movement expressed as "empty inside, strong outside," the lower nuclear trigram Ken refers to the arms/legs and the upper nuclear trigram K'an may refer to the heart or to the ear. Therefore, the motor interpretation of Chin can be as follows: (a) "sink slightly with soft knees" & "step left foot aside, in such a way that the weight ends up evenly balanced among the feet, the feet distance matching the shoulders distance, and the feet being parallel" (Figure 2B); (b) "let body and arms rise upward, the latter to shoulder height in front" & "slightly draw the wrists toward the shoulders, slightly straightening the fingers" (Figure 2C); (c) "continuing the circular movement of the arms, gently press them down to sides again" & "sink the body again" (Figure 2D). Second Movement - Code 111111 The name of the hexagram is Ch'ien which means the creative and Ch'len are also all the component trigrams. The meanings of the trigram Ch'ien include heaven, strong, moving forward, and turning. This form, after the smooth previous one, is "strong" and It consists of a turning phase and a forward moving phase: (a) "while shifting the weight to the left leg, turn right, pivoting on the right heel" & "bend the elbow until the open hand reaches the shoulder height with the palm Inside" & "raise the straight left arm in front of the body up to the waist level, palm up" (Figure 3A); (b) from the previously reached posture, "the body turns left to the initial direction" & "the left leg steps forward, slightly left" & "the left hand moves up/front at the chin level, palm facing the chin" & "the right hand returns to the right side" (In performing this movement, the weight must first shift to the right foot and then to the left foot); this movement pattern can be briefly named "grasp the bird's tail" because the left palm is held at chin level as though grasping the head of a big bird and the right hand moves to the side of the body as though smoothing the bird's tail (Figure 3B). Note that in the description above of T'ai Chi Ch'uan forms, the "" isolate movement units and the & connective stands for simultaneous performance, borrowing from the NEM terminology, which is developed in another chapter. As already stated above, infinite movement sequences can be generated. An interesting movement pattern, different from the canonical sequence of 108 movements, is the following sequence of 12 movements denoted by the following sequence of symbols: 000001 000011 000111 001111 011111 111111 111110 111100 111000 110000 100000 000000

Figure 2. First Movement of T'ai Chi Ch'uan

Dance and Movement Notation 95 This may be interpreted as a "waxing and waning of Ch'len (strength) and K'un (weakness)", i.e. as the smoothest cycle of energy build - up and release. Apart from philosophical or cosmological considerations, T'al Chi Ch'uan Is quite remarkable, In our opinion, for its modernity in formalizing aspects of human behavior (natural movements, in particular), which is the focus of contemporary Artificial Intelligence. Furthermore, it captures another central concept of motor knowledge: the complementarity between movement and shape. On the other hand, the T'ai Chi Ch'uan formalization is rather Imprecise at the low level of movement - • = - ntation and this leaves open the problem of Interfacing it with a sufficiently powerful low level formalism. / Figure 3. Second Movement of T'ai Chi Ch'uan

96 A. Camurri et al. 2.2. A Western Approach: the Eshkol - Wachman Notation The Eshkol-Wachman notation is the result of an attempt to formalize In a precise way the movements of the human body, with particular attention on the problems faced by a dance composer. The result is a "movement notation" and

not a "dance notation" in that It is not limited to a particular dance style and it aims at expressing every visually discernible movement of the human body. The EW notation system, started in the fifties in Israel as the joint effort of Noha Eshkol and Abraham Wachmann [Eshkol and Wachmann 1958], is perhaps the most rigorous attempt of the pre-computer era. Significant improvements may only come from the methods and the concepts of Artificial Intelligence which may provide the opportunity of rigorously expressing the quantitative geometric patterns underlying human movements (well investigated in the EW system) and the qualitative "linguistic" aspects which are well expressed for example by T'ai Chi Ch'uan or highly formalized dance styles. The ambiguity and richness of these aspects can be associated with the extreme richness and ambiguity of natural languages (both eastern and western) when dealing with space, spatial relations, and movement. In the EW notation the moving body is considered as an orchestra where every limb is a separate instrument. The human body is schematized as a system of 18 limbs (3 limbs for each leg, 4 limbs for each arm plus head, neck, torso, and pelvis) and two pseudo limbs are added (weight and frnnt) to account for the relation between the body and the environment: a total of 20 limbs. The EW notation is organized in a similar way to the music notation with "movement symbols" (analogous to the music notes) written on a 20 line diagram which is analogous to the musical pentagram (we may call it icosagram, from icosa»20). Each line corresponds to a limb and the time flows from left to right, discretized into uniform time intervals marked by vertical bars on the icosagram (Figure 4). The duration of the time quantum is arbitrary and it is specified at the beginning of a notated movement in terms of metronome settings. i: ' fH' I-and Fr«-m ^ wu'*tr Vy, 3im.lkr L»«l I ~ppr pjet I ' , * W I' 14"! I i i i i i i i i i i i i
Figure 4. Twenty Line Diagram of the Eshkol-Wachman Notation System

Dance and Movement Notation 97 Each line of the icosagram, which corresponds to a given limb, contains a sequence of movement symbols which correspond to elementary motor actions: movement symbols ::-^/V./-* /«-//T'/^ lulT~ These symbols, as better explained in the following, identify different types of joint rotations. Limb trajectories or other motor patterns cannot be expressed in the notation. The geometric aspects of each elementary motor action are represented by means of numeric attributes which specify extent, direction, and orientation of a movement. The numeric attributes imply well precise reference systems, one for each limb, and derive from a quantization process similar to the discretization of the time axis3. The size of the rotation quantum, as well as the time quantum, Is written at the beginning of the notation. The duration of a movement is denoted with a vertical heavy bar, placed one or more time slots at the right of the movement symbol (changing the position of the bar while keeping the numerical attributes of a movement specification, changes the movement speed). Every joint has a reference system of its own (its origin is rigidly fixed with the joint) plus one system for the body (Its origin Is located In the body center). These systems remain parallel for any body movement and they keep their orientation with respect to the horizontal plane except for oommon rotations around the vertical axis which occur during turning. As a consequence, these systems measure absolute, not relative rotations4. A spherical system of coordinates is used for each reference system. For example, in the reference system located at the elbow the location of the wrist is identified by two angles (similar to the longitude and latitude of geographic coordinates). These angles (which are quantized as explained above) can be used to express different things: position of one joint with respect to another, amplitude of a rotation, or direction of the axis of rotation. In the EW notation three types of primitive motions are considered: (a) Rotatory movements, Identified by two semicircular symbols; (b) Planar rotations, identified by four arrow - shaped symbols; (c) Curved conical rotations, identified by two corner - shaped symbols. Rotatory movements are limb rotations whose axis coincides with the limb axis. The two symbols distinguish the clockwise vs. the counterclockwise direction. A number (above or below the symbol) specifies the rotation angle. Plane movements are rotations of a limb with a rotation axis perpendicular to the limb axis. Three types of notation are used for (i) horizontal, CO vertical, and (III) oblique planes. In the first case the horizontal arrow is used (its direction distinguishes olockwlse va. counterclockwise rotations) together with a number (written above it) which identifies the rotation angle. In the second 3 The stated rationale lor the quantization ot the geometrical entitles such as the joint rotations or joint positions is that the accuracy ot perceiving and executing movements Is not very high. However, we wish to remark that, although 'accuracy* Is low. "resolution" is very high (one ot the many paradoxes ot biological Information processing). Furthermore, when considering movements constrained or directed by the environment, the geometric parameters ol the movements may have arbitrary values and cannot be dlscretized without losing significant information. ' The alternative would be to use relerence systems which are rigid with respect to the limbs and not rigid with respect to the environment. The two types ot representations are equivalent in the sense that it is always possible to trenslorm ona Into another. However, the absolute approach is better suited if we emphasize the "visual appearance* of movement whereas the relative approach is more closely related to 'motor control' Issues.

98 A Camurrietal. case the up-going or down-going arrows are used and the rotation angle Is denoted by a number on the right. Moreover, since there are Infinite vertical planes, it Is necessary to denote the actual plane by means of a number, written between brackets at the left of the symbol, that specifies the orientation of the plane (i.e. its longitude). In the case of oblique planes, for Identifying their orientations It Is necessary to specify one more number (the latitude, In addition to the iongitude, of the normal to the plane) which is written at the left of the arrow on the top of the other. Curved conical movements are limb rotations around an axis that forms an acute angle with the limb axis. (The other two types of movements are just special cases of conical movements and the distinction is done only for convenience.) The two symbols are used for distinguishing clockwise vs. counterclockwise rotations and the numerical parameters which are necessary have a similar meaning as in the plane movements on oblique planes. Two numbers (written In brackets on the left) denote the orientation of the rotation axis (longitude Is the bottom number and latitude is the top number) and another number (written on the right) denotes the rotation angle. Turns of the body are denoted using the same symbol of rotatory movements and a special line In the icosagram is provided (the first one) which records the changes of the front direction of the body. In addition to the movement symbols, the EW notation has symbols for denoting the contact with the ground and the distribution of weight over the support base. Any limb of the body can be in contact with the ground. Making and leaving contact are denoted, respectively, by a square and a circle and a parameter written Inside the symbol specifies the precise modality (e.g. In raising a foot from the ground it Is necessary to distinguish which part of the foot is being raised: the whole foot, the heel only, the front part only, the toes, the right aide or the left aide of the foot). An upper case L is used to denote "Loose Contact." The contact symbols are put In the appropriate line of the EW icosagram. The location of the weight with respect to the support base is stored In the second line of the icosagram. A filled circle Is used to denote the

fact that the weight is evenly distributed, i.e. the vertical line through the center of gravity of the body hits the ground in the centroid of the support base. In order to denote displacements of the weight from this situation, rotation symbols are used, with a slightly simplified syntax: In fact only vertical planar rotations and conical rotations with vertical rotation axis are significant. Finally, since an EW notated movement consists, for each limb, of a sequence of "relative* motion symbols which can be applied to any initial posture of the body (compatibly with the rotation limits of the joints), the notation needs a way to denote the Initial "absolute" configuration. This is done by defining a zero configuration of the body (erect vertical standing posture; arms and legs straight and parallel; feet parallel; palms toward the body; weight evenly distributed on the feet) and by denoting it with a zero. Initial postures different from the reference one are denoted by means of the same numerical notation used for example to specify the rotation axis of conical movements. A couple of numbers, one above the other between two brackets, represents the spherical coordinates of a limb. With such a convention, the zero position corresponds to zero spherical coordinates for all the limbs. The Initial body configuration is

Dance and Movement Notation 99 written between the first two vertioal bars of a notated movement. The table summarizes the syntax of the EW movement symbols. An example of a notatsd movement is given in the Figure 5. Rotatory Hovwnta VJ/ CCW : direction ^-^rotation angle rotation axia*li«b axLa r« Plana Hovtmnti 1. Horiiontal Plan* [CW CCW : direction ^—> rotation angle rotation axia*vertical 2. Vertical Plan* (2 it3 < 2)1*3 up down : direct ion •orientation of the vertical plane "' rotation angle rotation axis-horizontal 3. Oblique Plane down . direct ion orientation of the plane tilt the plana rotation angle ' rotation axis*sxewed Curved Conical Movement! CCW : direction spherical coordinates of the axis rotation angle Contact with the around a o Making contact Leaving contact Bight _•_. Center of Gravity Center of Support Table of EW Notation Symbols ajthmira rj" • 1 ° « « o 0 a T a I j. 1 0 T 1 * L' a -_- * >■• * <■• . r 1 * a M In * 0 • '4 , ■ i • (* M» 0 iV» h4l fttl I • 0 — ~li~ i.. • ■ j T • Figure 5. Example of the EW Notation System

100 A Camurrietal. 3. DANCE AND MUSIC Although sound and movement are obviously quite different from the physical and phenomenoiogical point of view, music and dance are strongly related human activities. A complete

discussion of this issue is clearly outside the scope of the book, but we wish to single out a few points that are relevant to our investigation of the computational bases of movement. For example, typical musical concepts such as rhythm, melody, and harmony can be applied to movement as well5. A link between music and movement is determined by the obvious fact that the execution of music is also a motor performance. Moreover, on a deeper level, there is a three-ways link between music, shape, and spatial aspects of movement: music is indeed a process which involves spatio-temporal structures as well as sound. The deep cognitive relevance of this link is revealed also by the abundance of terms, in music terminology, that are clearly borrowed from the vocabulary of movement and shape description (see the appendix on music terminology). Moreover, the two different aspects of movement that can be associated with music (motion skill of the performer vs. spatio-temporal structure of the performed music) mirror the two different methods of trajectory formation that we discussed in the previous chapter (intrinsic vs. extrinsic representation of a movement). On an even deeper level, we may think of improvisation, and in particular, of producing variations on a theme: this consists of two methods (repeating the same pattern with different parameters and/or overlapping it with concurrent patterns) that may be applied in a similar way to both music and movement. If we now shift our attention from music to music notation, we see that the situation is quite different: for music indeed, differently from dance, a successful notation method was found, which captures the essential structure of music and can be stored on a common medium such as paper. The adequateness of the music notation symbolism to express the essential structure of music (what in Al terms could be called music knowledge) is also proved by its ability to survive the advent of the computer era. Music notation can be easily expressed in computer terms and can be used directly to drive computer generated music, embedding it in the large domain of Al techniques6. For dance and movement, on the contrary, the picture is quite different. Dance notation methods have proliferated, without finding the same success of music notation and the computer techniques directly applied to them do not seem so far to pass the basic test: the ability to generate from the notation a fluent natural synthetic motor performance. We would like to emphasize this point because it characterizes our point of view about artificial intelligence: "We can say that we understand some aspect of human intelligence when our Al model is able to emulate it." Therefore we think that the ability to go from notation to animation is not simply a trick for computer hackers but an important epistemological and methodological goal. 6 The similarity between the Icosagram ol the EW movement notation and the pentagram of the music notation suggests lor example to regard body configurations as "movement chords". Moreover, the Interpolation in the motoric context ot the concept ot scale (and melody) is discussed by Lewis and BartenleH (see also section 5). 8 Not all the music can be written in the traditional notation which was born In the Renaissance. This Is certainly not the case ot some contemporary music (Scheefler, Cage. Berio, ...). A notation Is always an abstraction which leaves out something. Nonetheless a notation is succeslull il It can capture a large part of reality and Is sufficiently 'user friendly".

Dance and Movement Notation 101 Coming back to the issue of the similarities and dissimilarities between music notation and dance notation, we would like to point out the basic fact that In music notation one of the organizing principles is the time axis. Time is the independent variable along which the musical events are located, determining musical rhythmic and melodic structures. Subdivision of the time axis into intervals and subintervals creates the rhythmic structure. The use of the pentagram and its associated symbolism allows to specify concisely and unambiguously the attributes of the musical events which must happen into each time-slot. Therefore, taking into account this kind of discretized time axis we may say that music notation is event-oriented because it is organized as a fiow of musical events7. It is important to point out that the notation considers the events as discrete even if the physical musical instruments may be characterized by smooth continuous characteristics. However, this kind of structural complexity, which has to do with the physics of the particular instrument, is not represented in the notation. The Instrument itself is the appropriate representation of the unique cromatic and timbric features of the sound that it can generate and thus there is no need to make it explicit in the notation. Similarly, what is not (and cannot be) expressed in the notation is the particular way in which a performer plays the specific instruments and the individual style of the performance. Summing up, we may say that music notation is an example of a symbolic representation of a complex phenomenon which is successfully based on an organizing principle able to discriminate which aspects of complexity to represent explicitly and which ones to represent implicitly, leaving them to the cultural knowledge of the user (i.e. the composer and/or the performer). In comparing music and dance notation, the first question that we must answer is the following one: Can dance be represented as an event-oriented phenomenon? All the people who developed dance notation seem to imply the affirmative answer, since, in one form or another, these systems use a staff similar to the musical pentagram, with more or less lines , traced horizontally or vertically. There is, of course, the cogent reason that in many cases dance needs to be synchronized with music, but there is also the well accepted concept that a dance performance can be described in terms of "notes" and "chords" in a similar way to music. However, the real question is the following one: If we agree that dance can be represented as a stream of events on the discretized time axis, which are the events that we should single out and which are the relevant attributes that we should explicit? No unique answer to this question has been found so far. The main musical event, the note, has very nice features from the point of view of the power and simplicity of the representation: (a) It Is characterized by few parameters (pitch, duration, intensity, expression); (b) it has a clear epistemological meaning for the composer and the performer, i.e. it can be used for very different purposes (composing, learning, rehearsing, documenting, storing, etc.); 7 The "atomic" musical event is the note, but that is not all. For music indeed, as lor any other humeri activity which has a cognitive content, meaning and structures are recursive entities. Furthermore, the Mow of events is not necessarily synchronous but may depend on some kind of 'message passing*.

 (c) it is sufficiently abstract, i.e. independent of the peculiarities of particular Instruments or performing techniques; (d) it is a simple primitive entity from which it is natural to build complex macro structures according to a set of linguistic rules. Which aspects of movement are able to play the same fundamental role of the note? The answers can be quite different, as it is very well reported in the book by Ann Hutchinson [Hutchinson 1984]. In general, we may outline a wide spectrum Indexed by the degree of detail of the notation. At one end of the spectrum we may have very gross notations, characterized by macroscopic representations, and at the other end we may have very detailed notations which specify such fine phenomena as individual rotations of the Joints or Individual muscle contractions. The Renaissance notation system by Domenico da Piacenza and his students is an example of a concise/macroscopic notation. It uses only five symbols, which correepond to five basic movement patterns used to compose the dances of that period: R ("reverenza": the starting bow), s ("sempllce": a step forward followed by closing the feet), d fdoppio": three step forward followed by closing the feet), b ("branle": a swaying step), r ("ripresa": a backward step). The representation of the dance is organized as a monodlmenslonal stream of symbols, e.g. "R b ss ddd ss rrr...." . These symbols are linked to the musical line, In a similar way to the words of a song. The plus of the notation is Its simplicity and conciseness, but Its obvious drawback is its Inability to represent other movement patterns than those Implied by the limited and specialized set of macroscopic motion primitives. The dissatisfaction with this kind of drawback has motivated subsequent inventors of notation systems to Increase the degree of detail captured by the notation and we can in fact read the history of dance notation systems as an evolution from low to high detail. An example of this kind of trend Is given by the modern system invented by Eshkol and Wachman, where all the joints are explicitly represented by individual lines of a large staff. The human body is viewed as a stiok figure, composed of links and joints, and the notation provides the symbols for specifying the rotations of each link for each time slot. However, this is not the ultimate level of detail. We may even go farther in describing the microscopic structure of movement. For example, we might represent the Individual muscle contractions or the activity patterns of the motor units of each musole, and so on. On the other hand, in a large spectrum of formalized motor activities ranging from dance to artistic and rhythmic gymnastics, natural language is still the description tool most widely used, complemented by the aid of specialized lexicons and of image sequences. Certainly, imprecision, ambiguity, and lack of standarlzation characterize this method of description but it is also true that it embodies a large amount of common sense knowledge, and this makes It very attractive to

Dance and Movement Notation 103 Artificial Intelligence. With regard to the eidetic part of informal movement description, Figure 6 shows, as an example, the image sequence of an intermediate level competition floor exercise in artistic gymnastics8. It is Interesting to note that the sequence of images does not correspond to a uniform sampling of the time axis. On the contrary, the images spot body configurations, either static or It J ■ f I I < « Figure 6. Free Body Exercise in Artistic Gymnastics " Thia was (he obligatory lloor routine (Iree body exercise) for girls ot the A3 category In the 1984/1185 sporting year of the Italian Gymnastics Federation.

 dynamic, which characterize each movement pattern and which can be object of a quantized judgment. It is also possible to argue that this kind of adaptive behavior (choosing the sequence of key Irames9) is an efficient code, which captures the essential aspects of complex movement patterns according to experience and insight, while disregarding many unnecessary details. The latter point is also confirmed by the fact that sequences of images from a movie (sampled at a uniform rate) look much more static than sequences of drawings which sample the time axis in a non uniform, "intelligent" way. It is also interesting to note that in Figure 6 the number of frames is 50 and the execution time of the routine is less than two minutes, which may be interpreted as a code production rate of two to three decisions per second. The visual code, although efficient, lacks the combinatorial and generalizing power of language, which is indeed used inseparably with visual descriptions such as that of Figure 6. Natural language provides a large dictionary of space and movement related terms which corresponds, in terms of Artificial Intelligence research, to spatial and motor knowledge. Similarly, special motor knowledge has been coded in specialized jargons or lexicons, where each term (e.g. "jete'" or "cartwheel") may be considered as macros or subprograms of a motor language. In Appendix B we show an elementary spatial and motor lexicon which we developed by collecting the terms used in several books for describing movements or sequences of movements [Esh- kol and Wachman 1958, Hughes 1983, Liu 1973, Schmid 1976, Bartenleff and Lewis 1980, Hutchinson 1984].

4. CONCURRENCY AND MULTIPLE HIERARCHICAL LEVELS

When representing movement patterns in an increasingly detailed fashion, in the end we might be content only with a complete replica of all the neuromuscular events which compose an action. However, the pursuit of finer and finer uitrastructures, so successful! in nuclear physics, is likely to be en epistemological dead end in the study of intelligence and purposive behavior. In natural as well as artificial vision, David Marr emphasized the crucial importance of dealing simultaneously with course and fine representations, because each of them, by itself, fails in important aspects. In the field of movement, the same idea is directly applicable. For representing a certain movement pattern it may be convenient to single out quite different types of events at very different levels: 1. muscular level, 2. joint level, 3. spatial level (trajectories of limbs in space, etc.), 4. environmental level (correlation of the movement shape with the object shape), • Note that this kind ol coding is typical of comic strips and ol animated movies. The comic strip author only draws the key Irames, leaving the task ol re-creating the continuity ol action to the imagination ol the reader (i.e. to his implicit knowledge of the world and his ability to Interpolate) In animated movies, the common technique is to drew first the key Irames and to interpolate afterwards, drawing the necessary sequence ol Images.

Dance and Movement Notation 105 5. physical level (gravity, constraint, support, etc.), 6. action level (walking, jumping, etc.). In other words, we need both a macroscopic notation such as that of Domenico da Piacenza and a microscopic notation such as that of Eshkol - Wachman, but we must combine them in a dynamic way and this is not compatible with any of the fixed types of staff proposed so far. The multiple level, distributed type of representation that we are trying to outline has another puzzling aspect. If we think of the stick figure as our target model whose "natural movements" we are trying to represent, the distributed paradigm of control implies that the rotations of the individual joints of the stick figure are the effect of the superposition of multiple, simultaneous processes operating at different levels. This kind of functional topology is very well known in motor neurophysiology where the alfa motor - neurons are the "final common pathway" for a large family of spinal and supraspinal motor centers, which carry the results of quite different types of computations. Therefore, the output of the motor system, I.e. the resulting set of joint rotations and/or of muscle contractions is the worst locus where to center a motor notation system because extremely complex output patterns may result from the interactions of even simple and regular motor processes acting at different levels in the hierarchy of the motor system. Walking is a very good example where an extremely complex superficial structure changes in a subtle way as a function of walking style, walking conditions, changes of rhythm, etc. We ere in a better position now to try to answer the question that we formulated before: Which kind of motor events should we explicate on the time axis in order to synchronize them with non motor rythmic events such as music? The answer is that significant motor events can be very different things: actions or motor processes at different levels of granularity such as rotations of the joints, trajectories in space of different parts of the body, oomplex patterns such as reaching and jumping, etc. No one of such actions is necessarily observable because the actual movement pattern is the superposition of many of them acting simultaneously. It emerges from a "pandemonium" of underlying little "motor demons." The notion of pandemonium as a metaphorical model of the information processing performed by the brain, is due to Selfrldge [see Lindsay and Norman 1977] who proposed it in relation to visual pattern recognition. This system is composed of an ensemble of demons who work on the pattern, each performing a different job (see Figure 7). The first set of demons, the image demons, merely records the initial Image of the visual stimulus. The image is next analyzed by the feature demons, each of them looking for a particular characteristic in the pattern: the presence of a certain type of line, of angles of some sort, etc.. Cognitive demons watch the response of feature demons. Each of them is responsible for the recognition of one pattern. Thus, one would try to recognize 'A', one 'B' and so forth. For example, the A cognitive demon tries to find evidence for the presence of features associated with the 'A' pattern. The more features it finds, the louder It yells. A decision demon listens to the "pandemonium" produced by the population of demons and it selects the pattern which Is yelled by the loudest demon. The pandemonium concept can be used to model many different types of brain activity, and in particular motor planning and control. In thla kind of pandemonium we shall find Joint demons, trajectory demons, spatial demons, gravity demons, skill demons, etc.

106 A. Camurri et al. Figure 7. The Pandemonium The - • .; tation of a given movement as a stream of Interacting motor • >- -- ■ (i.e. demons) is obviously not unique. Many different ■■>> ■ -ations yield the same result, as many different computer programs can carry out the same computation. But this Is a cognitive plus: it gives the opportunity to use more or less macroscopic representations according to the context. For example, it may be useful to go from coarse to fine representations during learning, rehearsing, or teaching and the other way around In skilled articulated behavior. In . , the possibility of transforming a representation Into another one is likely to be a very powerful feature. In any case, for any selected context, our motor pattern can be ; • =. • as a stream of actions. How are we going to synchronize them either mutually or with respect to external events, such as music patterns? Three basic types of synchronization among actions can be taken Into account which mirror related ►• ►•In robotics: (a) sequential (perform action 1, then action 2,... then action n, one after the completion of the other); (b) parallel (perform action 1, and action 2,... and action n, at the same time) (c) guarded (while condition perform action; wait until condition, then perform action;...) The first two are synchronous ways to combine actions, whereas the third one reflects different

types of asynchronous paradigms. A complex action pattern can be decomposed Into simpler actions, joined with synchronous or asynchronous operators, and such action can be decom ••>-•, In turns, into other action patterns, until primitive motions are - •, such as

Dance and Movement Notation 107 those used in the Eshkol - Wachman notation or in NEM. 5. MOVEMENT AND SHAPE Movement and shape are mutually related: Shape determines movement and movement determines shape. The former statement means that it is the shape of objects - their geometric structures - which allow to plan purposeful movements. This concept Is present in T'ai Chi Ch'uan where many movement patterns are expressed in terms of shapes or geometric features of natural objects. In a more formalized way, the same idea has been followed by Bartenieff and Lewis [Bartenieff and Lewis 1980] in the framework of the Laban movement notation [Laban 1956] . They consider the sphere which surrounds a human body and they call it "kinesphere," a sort of generalization of the well known anthropometric circle by Leonardo da Vinci; movement patterns can be described inside the kinesphere and, in particular, sequences of movements which are aimed at the vertices of solid figures inscribed inside. For example, Figure 8-top shows a sequence to the 12 vertices of an icosahedron (a regular polyhedron with 20 faces). The analogy with the cromatic musical scale is obvious, and indeed the authors speak in terms of "movement scales." In general, this is an example of the way In which shape can influence movement. The dual aspect can be considered by focusing on the shapes generated by a moving body. This point has been studied for example by Harries [Harries 1983] In the context of the Eshkol - Wachman notation. Figure 8-bottom shows an example of a trajectory generated by a simple two-link arm (3 and 2 are the lengths of the links) as effected by the simple EW program written in the figure. This kind of constructive representation of shape Is embodied in programming languages like LOGO. Furthermore, as remarked also by Harries, it is possible to extend the idea by considering an arm trajectory as the medial axis or skeleton of a shape. This point, in particular, is treated at large in another chapter of this book. Another aspect of movement notation, which is relevant for the visual appearance of a performance is the "virtual" movement of the observer which is made possible by filming or computer techniques. It is clear indeed that the movement patterns of the real or simulated camera are not "neutral" but tend to influence substantially the action. In video dance or computer dance this means that the camera movements become an integral part of the choreography, as if the camera were a sort of virtual dancer. That means that the formalization and notation of the camera movements can be approached similarly to the dancer movements, only taking into account the particular motor capabilities of a camera: (a) orientation of the camera in space, which implies three independent rotations (the namee commonly used among filming people are "panning" for rotations around a vertical axis, "tilting" for rotations around an horizontal axis orthogonal to the optic axis, and "cartwheeling" for rotations around the optic axis. These are the familiar Euler angles.);

 Isl0» (27)|16 (27) f 6 Figure 8. Shape and Movement - Movement and Shape (b) translation of the camera In space, which implies three more degrees of freedom, with special names being used for particular trajectories (e.g. "trucking" to denote lateral translations, "dollying" to denote translations toward a subject or away from a subject, "circling" to denote movements on a circular path, while keeping the view axis on the same subject); (c) virtual movements due to the optics ("zooming in", "zooming out", changing the plane of focus). Attempts to modify current dance notation techniques to take Into account camera movements have been made. For example, Mahoney [1964] proposed an extension of the Labanota- tion in which the camera has its own staff, in addition to the dancer staff, with special symbols for the different types of camera movements. In the field of computer graphics and geometric modeling, a common feature allows choosing arbitrarily the point of view and the optical characteristics of the "synthetic camera." However, for the camera movements as well as for the human movement, recording the movement is quite a different thing than "understanding" it. There are still fundamental cognitive aspects which need to be faced. 6. REPRESENTATION AND EIDETIC PRESENTATION OF ACTIONS In summary, an action pattern is, from a logic point of view, a graph in which the main action Is decomposed Into a network of subactions: an action graph. The subactions are implemented by "actors" or "demons" that can exchange messages. The action graph is dynamic: parts of it may be generated or deleted during the actual performance, as a result of "sensory

 feedback* or "knowledge of results." An important issue in action representation is the relation between the dynamic action graph and the knowledge data base from which "instances" of "generic" nodes may be produced, perhaps in multiple interacting copies. This problem is discussed at length in the next chapter. The action graph is logically coupled to the model of the actor as a sort of "cognitive shadow": each node of the graph determines a modification in the configuration of the model. This is a many to many relationship because each node of the graph can affect several joints of the body and each joint can be affected by different nodes, either acting simultaneously or not. (This complicated topology is sketched in Figure 9.) It is important to point out that even for apparently very simple actions of everyday life, the corresponding action graph can be extremely complex, making It difficult to generate (i.e. programming the action) or to visualize it during performance. The actual performance is a process whereby the nodes of the graph are activated/disactivated and signals/messages are exchanged according to very complex timing patterns. How can we represent such a process? Of course, there are infinite ways (remember the metaphor of the Japanese garden of the initial section) which consist of selecting elements of the action graph and visualizing them. Also such eidetic phase is far from unique. Let us consider the following examples: (a) We can use a time - oriented presentation, similar to the music notation, by putting the selected kinds of events on some kind of staff; (b) we can display the global output of the action by using computer graphics and/or computer animation techniques or by filming the actual performance of a dancer and we can choose the real or virtual point of view according to the visual element that we wish to emphasize; Figure 9. Action Graph and Body Skeleton

 (c) we can single out the special kind of correlation that exists among two or more variables by visualizing the trajectories described by them in the corresponding space. (For example, it may be interesting to look at the butterfly - shaped trajectory described by the center of gravity on the coronal plane of the body during walking.) These three paradigms correspond to generalized types of eidetic presentation of the performance of a process: (I) time course of state variables, (ii) visual simulation, (iii) phase plane trajectories of state variables. Therefore, we have a wealth of useful alternatives in representing the performance of an action. We must decide which aspect to look at and how to look at. The computational approach to movement notation that has been discussed above leads naturally to the project of a computerized environment for the interactive representation of motor patterns. The NEM language, which is described in another chapter, Is an initial step In this direction. In this programming environment one should have available data bases of (i) postures, (ii) actions, (iii) objects, and (iv) musics and should be allowed to enrich it by editing old or new patterns. Starting from such data bases it should be possible to write composite action programs and to synchronize them with music by means of constructs of the type "at beat # X perform action Y." Moreover, one should be allowed to select an eidetic interface which shows explicitly some spatio-temporal aspects of the action during its performance - even elements which were not directly mentioned in the action program. Finally, let us come back again to the music/movement comparison. Although the music notation has been much more successful than any dance notation system, many of the structural concepts and subtle complexities of movement are equally valid in music, for example the concept of multiple level, distributed type of representation that we discussed above. Indeed, the representation and description of musical processes in terms of Petri Nets, developed In another chapter of this book, is perfectly compatible with the general Ideas of movement representation discussed in this chapter and the computer techniques of movement formalization presented In the chapter on the NEM language. The point Is that all the sources of complexity of musical structures are significant aleo for movement structures. In the latter case, however, there is an additional complexity - geometric structure. Apparently it Is easier for human beings to master a low level representation of musical structures than of geometric structures. On the other hand, it Is not clear to what extent this inadequacy is subjective (i.e. related to 'cognitive bottlenecks" of human beings) and to what extent it is objective (I.e. related to loss of key information).

Dance and Movement Notation 111 APPENDIX A: ELEMENTARY MUSIC NOTATION AND TERMINOLOGY Lat us summarize some basic musical concepts which are significant from the point of view of dance and modern gymnastics (we restrict ourselves only to western post-Renaissance music10). In particular, let us consider rhythm, melody, harmony, agoglca and dynamics. Rhythm expresses the relationship between the sound flow and time. In analyzing the structure of music, it is possible to find a series of grouped pulsations or beats. The beat may be even or uneven, fast or slow. Each recurring group of beats is called a measure. The measures are separated from each other in music notation by vertical lines or bars. Accent can be described as additional force placed on certain beats in a measure. Most dance steps are accented on the first beat. Melody refers to the "heigth" of sound (the pitch), its evolution in time and its relationship with measures. Melody is often said to be the "horizontal" dimension of music, since it copes with a single valued function of time (pitch vs. rhythm). A note is a single sound characterized by pitch, duration and other attributes: different symbols are used to specify duration, whereas pitch is expressed by the position of the note on the pentagram. A phrase is a sequence of notes which expresses an "Idea" i.e. a structure on a meta- level with regard to music notation. At this level we may think of phrases as "musical sentences" which, as poems, are divided by melodic and rhythmic punctuations. Often, phrases contain four, eight, or sixteen measures. At a higher level of abstraction, the term musical form is used to define the architecture of a musical piece (motet, madrigale, ballet, etc.). The number of beats in a measure is called the meter. There are two important types of meters: duple (with two beats to each measure) and triple (with three beats). Almost all other meters are multiples or combinations of these. For example, a four-beat group (quadruple meter) consists of two groups of two beats. Distribution of accents inside a measure gives a strong characterization to a rhythm: The consequence is that two rhythms may sound very different although having the same meter. A meter may contain notes of different durations, where one beat is the unit of duration: (i) A "whole note" lasts 4 beats, (II) a half note lasts 2 beats, (iii) a quarter note lasts 1 beat, (iv) an eighth note lasts 1/2 beat, (Iv) a sixteenth note lasts 1/4 beat, etc. A further timing opportunity is provided by the "dot" symbol which, applied to a note, increases its duration by half of its value. The time signature is used to show the metric division of a measure. For example, a time signature of 3/4 means that each measure will have three beats and that each beat will have a quarter note. A piece of music In 3/4 meter need not have the same rhythm of three quarter notes in every measure but the notes in each measure must add up to a total value of three quarter notes. Consider, for example the following different rhythmic patterns, all in the 3/4 meter: (i) (quarter quarter quarter), (ii) (quarter eighth eighth quarter), (Hi) (quarter-dot eighth quarter). 10 One of the anclenl notations which proceeded the present ona Is the chlronomlc notation (approx. until the 9th century). This is interesting from our point ot view because it used symbols ("neuma") which mimicked the hand movements ot the chorus master.

 Tempo refers to the speed of the music. It may vary from fast to slow. Conventional key words to denote it are 'vivace', 'allegretto', 'moderate*', 'andante', or 'lento'. Besides the "horizontal" dimension, music has a "vertical" characteristic, which is the concurrency among different melodic lines. Harmony is the set of rules which allow or prohibit certain combinations of sounds. These rules are strongly dependent on musical ages and cultures. Musical actions are defined by rhythm, melody, harmony, and synchronization among players mirroring the structure of dance actions. A particular interpretation of an expressive action requires some attributes: In music they refer to both rhythm (agogica) and intensity (dynamics). The former attributes control local variations on the rhythmic scheme: typical keywords of musical notation are 'accelerando', 'diminuendo', 'perdendosi', 'a piacere' and so on. Dynamics controls Intensity, that is the "force" of the music. It may vary from loud to soft. Conventional key words to denote it are 'fortissimo', 'forte', 'mezzo forte', 'piano', Rhythm Is a concept that applies to movement as well as to music and it was even suggested that human movement has a rhythm of its own. Rudolf Bode, one of the pioneers of rhythmic gymnastics, observed that the principle of moving to a rhythm is found in natural rhythms within the body such as the pulse rate and the respiratory rate. Music notation, although in endless evolution, is well established since several centuries. A whole dictionary of musical terms could easily fill thousands of pages. However, we think useful to recall some examples of terms that enlighten the analogy between music and movement. In particular, let us begin with a general terminology, thinking of music as a motor activity, we can use this simple classification scheme: 1. actions 1.1 [related to the] instrument 1.2 [related to parts of the] body 2. action attributes 2.1 body/space 2.2 time 2.3 force 2.4 shape 2.5 qualitative In Appendix B a similar scheme for dance terminology Is Introduced. The reader is encouraged to compare them in the framework of motor activities. The distinction between "action" and "action attributes" is basically the same as the distinction used in computer science between a "procedure" and its "arguments" or the distinction in mathematics between a function and Its arguments. For uniformity, the actions are denoted by the syntactic category of verbs, but this Is Irrelevant; however, with regard to action attributes we did not distinguish between nouns, adjectives, adverbs, and prepositions. Playing music is a specialized motor activity, characterized by high speed and skill for few parts of the body; the activity itself is bound to a limited "workspace" - between the performer and the Instrument - hence "actions" are determined by two goals'1: obtaining the proper sequences of notes (1.1), and moving parts of the body to find optimal postures and gestures for the performance (1.2). The terminology for the first kind of actions includes the standard musical

 notation, whereas the terminology for the second is currently used in teaching and training. 1.Actions 1.1 Instrument Actions on a particular instrument are mainly determined by the musical score. Its symbols evoke specific motor commands. For instance, a sequence of notes, say, G-A-C, may be considered a direct command to press specific piano keys with the proper attributes of timing, intensity, gesture and "shape" of movement. (The attributes are partially written in the score, and partially learned during training.) Actions refer mainly to: single notes pedals pauses chords registers percussions The movements which must be performed when playing an instrument are determined obviously by both notation and the instrument itself. However, there are some "motor/spatial concepts" which remain valid for any instrument. The same notion of "musical scale" comes from the latin "scala" which means staircase: Going up and down the scale is a dear metaphor that everybody understands12. Besides the symbolic level of musical notation (notes and pantagram), a large musical terminology exists which refers to more abstract musical structures ("themes", "phrases* and so on). Actions at this levels produce new musical structures characterized by the conservation of a particular musical aspect. Let us consider, for instance, "transposition": It transforms a musical structure into a new one which preserves the same relative pitch intervals, but requires a different motor structure when performed on an instrument. This resembles many motor paradigms, like handwriting: The same word, "transposed" in different spatial positions, preserves the same spatial structures although it entails very different movements. In dance and music, a same score can be performed with different interpretations. In music in particular, "interpretation" stands on an appropriate and creative use of "action attributes". Some examples of structural level actions are: trasposizione inversions aggravamento retrogradazione dilatazione ornamento 1.2 Body These actions specify a "gestual activity" that occurs during the execution of the score. They range from finger/hand precise movements to suitable movements of the whole body and the instrument. The related terminology is not usually noted in the scores (except for some special indications, like fingering). A musician, during training, learns gestual patterns, their techniques, and how to associate them to a musical context. Some examples of typical gesture activities (besides fingering) are: 11 On the contrary, the 'workspace' ol a dancer is much larger and may include objects as well as other actors. 12 This metaphor can ba seen at a higher level: the voices forming a musical piece lead the hearer to move along paths In an imaginary space. For instance, the term *lugue" (Irom the latin fuga, which means Might) is the name of a mu-

114 A Camurri et al. resplro (breath) alzata (raising, esp. of the hand) roteare (round movement of a finger) spostare (shifting, esp. of the trunk) appoggiare (laying, esp. of the hand) articolare (emphasizing finger movements) 2. Attributes Rhythm is a key point in music. Musical movements may reach a very high frequency. An artis for instance, can easily play at a rate of 20 notes per second for a long period of time. This fad joint to the great sensibility of the human ear, enhances the effects of even small variations c the timing of a musical performance and gives 'agogica'13, together with dynamics, a fundamer tal position in the basic set of attributes. Dynamics controls the force generated by the playe when interacting with its instrument. Force is an explicit attribute in music, unlike in dance. Thi stems from the fact that the player always interacts with a "compliant" environment - the instru ment - whereas a dancer does not. 2.1 Body/Space left hand right foot down bow close stick (These attributes are specific of body actions and relate either to a part of the body or to a spa tial relation between the body and the musical instrument.) 2.2 Time accelerando affrettando allargando a tempo calando senza tempo rinforzando trattenendo rallentando stentato animato calmando ritardando mosso stringendo con slanclo 2.3 Force ,4 tenuto forte con slancio violento crescendo mezzo forte energico lieve rinforzando diminuendo leggero marcato piano animato impetuoso scorn parendo 2.4 Shape These attributes synthetically express the evolution of an action from a "starting point" and i "final point". Musical actions need to be interpolated, at various levels, by the performer: whei a score specifies a sequence of notes, the "shape" of the execution may be left un - determined 13 A goad dellnition ol agogica was given by H. Riemann |Riemann 1884): 'Agogica Is the who* ol motor impulses regardless ot the prefixed rhythmic scheme, that a musical period requires per sa." 14 In some cases a term indicates both time and force as "con slancio"

Dance and Movement Notation 115 (Notes can be played with clear separation, partially overlapping and so on.) Shape attributes classify different kinds of required interpolations, although leaving a certain degree of freedom to the performer. In other cases, shape attributes are a sort of "macro specifications" for a pattern linked to one note in the score ftrillo', 'mordents' and, in general, 'grace notes'). pizzicato legato staccato appoggiato tremolo glissando trillo portato mordente acciaccatura appoggiatura gruppetto arpeggiato cascata cluster 2.5 Qualitative brillante agitato delicatoespressivo ripieno vivace ridestandosi con grazia etereo misterioso lusingando teneramente dolente con sentimento spiritoso divisi (These attributes describe, often in a metaphorical way, various characteristics of a performance.) Specialized dictionaries are used to integrate the general terminology with terms referring to a particular domain. In violin scores, for example, the following terms are detailed attributes, related to the movement of the bow, called "colpi d'arco" (literally bow strokes), alia corda gettato legato martellato picchettato staccato pizzicato portato tremolo saltellato col legno sciolto glissando

 APPENDIX B: ELEMENTARY DANCE AND MOVEMENT TERMINOLOGY We divide this small dictionary into two parts: a general movement dictionary and a special movement dictionary. The former one lists entries from natural language and the latter one from special movement jargons (which include many French terms from the French school of ballet, initiated in the seventeenth century). In both cases, the goal is semantic classification, without any attention payed at syntactic aspects. General Movement Dictionary The simple classification scheme that we use is the following one: 1. actions 1.1 [mainly driven by) gravity 1.2 [programmed with the purpose of] locomotion 1.3 [related to parts of the] body 1.4 [related to objects and geometric features of the] environment 2. action attributes 2.1 space 2.2 time 2.3 body 2.4 shape 2.5 qualitative aspects The distinction between "action" and "action attributes" is basically the same as the distinction used In computer science between a "procedure" and Its "arguments" or the distinction in mathematics between a function and its arguments. For uniformity, the actions are denoted by the syntactic category of verbs, but this is irrelevant; however, with regard to the action attributes we did not distinguish between nouns, adjectives, adverbs, and prepositions. Note that some terms may appear in different categories. For example, "bending" may refer to a part of the body but also to an object In the environment, whereas "supinatlng" is restricted to the body and "walking* Is just an example of locomotion. 1. Actions 1.1 Gravity rise fall tumble jump sit tilt raise sink stumble stand lay flip lift sag land hop support snap swing droop leap kneel suspend spin

Dance and Movement Notation 117 1.2 Locomotion walk jump 1.3 Body flex bow lean 1.4 Environment face place hold push 2. Attributes 2.1 Space close short in front down sideways below outside outward ascending from 2.2 Time slow speeding-up when lagging 2.3 Body head arm thigh shoulder knee upright 2.4 Shape straight arched hexagonal planar step leap extend pronate twist slide pivot release puil far long left forward peripheral behind on parallel descending at fast run hop stretch supinate tuck swing touch bring bend high vertical right backward central beyond under upright beside periodic slowing-down before while initial neck forearm shank elbow ankle round triangular corner symmetric simultaneous final trunk hand foot wrist supine convex square edge asymmetric jog turn rotate bend turn point kick low horizontal up overhead above inside inward turned over to rhythmic after leading subsequent pelvis fingers leg hip prone concave pentagonal face parallel

118 A. Camurri et al. 2.5 Qualitative Aspects gentle strong slippery rough smooth sudden small large loose heavy little slight firm light much Specialized Movement Dictionary The classification is organized as follows: 1. Positions and Poses 2. Body Movements 3. Steps 4. Jumps 5. Leaps 6. Turns 7. Rolls 8. Acrobatic Tumbling and Stunting 1. Positions and Poses Ballet Positions - These are five positions for the feet and five positions for the arms (see Figure 10) which may be used as components of other formalized actions. 1 *S^x=3? Figure 10. Ballet Positions

Dance and Movement Notation 119 Split - This is a position of the legs where both legs are extended and form an angle of 180 degrees. The split may be frontal or lateral and it may be ueed in a variety of patterns, either static or dynamic. If the angle is smaller, the term which denotes the same type of pattern is straddle. Plie' - Knees and hips are rotated in such a way that the center of gravity is lowered. In a demi-plie' the knee rotation is about 60 degrees and in a grand-plie' it is about 120 degrees. A sort of plie' is the "tucked" posture which Is assumed in some acrobatic movement, where the rotation of the hip is such that the knees come as close as possible to the chest. This is a posture characterized by a minimum moment of inertia which is essential for fast spinning movements of the body In acrobatics and in diving. Releve' - It consists of rising to the ball of the foot with extended legs. For example, a common exercise In the preparation of jumping is a series of "pile' releve' plie' releve' ..." starting from a first position of the feet. Ballet Point (or Ballet Curtsy) - This is a balance on two feet which may be reached from the fourth feet position by shilling the weight on the backward foot to a demi - plie' in such a way that the forward leg is stretched. The arms are In second position and the trunk is twisted and leaned in such a way that the arms are parallel to the forward leg. Lunge - Lunges are balances on two feet where the supporting leg is bent and the other ie extended, either to the rear or to the side. Arabesque - This is a balance on one foot with the other leg extended to the rear. It may be executed with a variety of arms poses, with different angles of the extended leg with respect to the horizontal and with or without a pile' of the supporting leg. Croise' - This is a balance on one foot in which the supporting leg is extended and vertical, the other leg is bent and forward, the trunk Is twisted toward the bent leg (croise'«crossed), and the arms are usually in fourth position. Altitude - It is similar to the croise with the difference that the bent leg is backward and rotated externally, in such a way to stay on a horizontal plane. 2. Body Movements Bending Movements of the Trunk - They may be executed forward, backward, or sideways. The forward bending can be done either standing or sitting and, in the latter case, the legs can either be extended forward or sideways (In a split position). The backward and sideway bending can be done either in a standing or kneeling position. In the trunk backbond, the final posture is the so called bridge, which Is the kernel for basic acrobatic patterns, like the walkovers.

 Circling and Twisting Movements of the Trunk - They can be executed with different angle of trunk bending, while standing or sitting. Waves - Waves are co-ordinated movements of the arms, the trunk, and the legs which "mimic" a wave. They can be executed in the sagittal plane (either forward or backward) and in the coronal plane. Both standing and knesling positions are possible. 3. Steps Waking Step - In dance and gymnastics it is characterized by extended legs. Running Step - In dance and gymnastics there may be different styles which emphasize some formal features of the motor pattern differently from the athletics context in which speed and/or energy are the main concerns. Hopping - It is a spring from one foot landing on the same foot. Skipping - It is a fast sequence of step - hop with alternate feet (it results in an uneven rhythm). Slide (also Glissade, Chasse) - It is a step in which one leg "chases" the other out of its position. It can be performed in all directions and It has the same rhythm as skipping. It is commonly used in preparation of other steps or jumps. Gallop - It is an exaggerated slide, a sequence of step-closa-step-close ... with the same leading foot. 4. Jumps Jumps are all characterized by the fact that the landing is done on two feet whereas in leaps the landing is always on one foot. The start in jumps may be on one foot or two feet, whereas in leaps it is always on one foot. In most cases, starting and landing is characterized by a demi-plie'. Saute' en Premiere - This is a vertical jump from first position to first position. Changement - This is a vertical jump from a fifth position to the complementary fifth position (e.g. forward right vs. forward left) with a crossing of the legs in the air. Entrechat - It is a generalization of the former with several rapid crossings of the feet In the air.

 Swan Jump - This is a jump where the body assumes the shape of an arch in the sagittal plane. Tuck Jump - This is a jump where the body assumes a "tuck posture." Cat Jump - It is a jump where the knees are bent and turned sideways in such a way that the legs form a diamond in the coronal plane. Straddle Jump - It is a jump with the legs extended sideways in a split. Cabriole - It is a jump where the feet beat in the air either forward, backward or sideways. 5. Leaps Ciseaux - This is a kick with extended legs which continues in a leap (scissor kick). Pas de Chat - This is a "cat leap" i.e. a scissor kick with bent knees. Jete' - It is a leap to a frontal straddle posture. It may be preceded by walking or running steps. Grand Jete' - It is a "jete" which reaches a full split In mid air. En/ambee' - In this leap two rapid alternations of a straddle or split pose are reached in mid air. Stag Leap - This is a leap which first reaches a "stag pose" (the forward leg is plie and the other is extended backward) and then evolves to a grand jete'. 8. Turns During turns, the orientation of the sagittal plane of the body is changed with respect to the environment. The rotation may be on two fset, on one foot or may occur in the air. The preparation of turns may consist of a sequence "step, plie', releve'" or simply "step, releve'." The direction of the rotation may be inward or outward with respect to the non - supporting leg. The size is usually a multiple of 180 degrees. During turning, the head is always the last to move and the first to arrive (a pattern known as "spotting" which requires to use a fixation point and has the purpose to reduce to a minimum the activation of vestibulo- ocular reflexes). Chaine' - These are turns on two feet which are performed while stepping sideways, a

122 A. Camurri et al. half turn for each step. The arms are usually opened to the second position during turning (to aid balance) and are brought to the first position on completion of the turns. Pirouette - The pirouette is a rapid spinning done on one foot. The free leg may be held in various positions (in demi-plie', in arabesque, In attitude, etc.). The arms are joined in first position during turning (to reduce the moment of inertia) and are opened to the second position to stop. From the initial posture (feet In fifth and arms in first) the spinning action is started by a battement tendu, a plie' and a releve' together with a movement of the arms which go from first to second and then to first again (the hand in the direction of turning before the other). Pique - This is a pirouette in which one steps with a straight leg onto the ball of the foot and completes the turn before returning to a plie' position. Tour Jete' - This turn is performed in the air after an Initial kick (jete), with the arms circling overhead. Knee Turn - Starting from an asymmetric kneeling (one leg bent to the front with toes on the floor) the two kneels are joined during turning and the same initial position is regained at the end. Split Turn - It is also called back spin. From a frontal split on the floor, the backward leg is swung forward and up to an inverted split position (while laying on the back) which allows to make a complete turn on the upper part of the back, The turn is finished by tucking the legs and coming to a kneeling position. 7. Rolis While turns are rotations of the body around a vertical axis, rolls are rotations around a horizontal, transversal axis. The support of the body weight is transferred from the legs to the back and then to the legs again. The rolls may be performed without the aid of the hands, or with them; with extended or tucked legs, with symmetric or asymmetric position of the legs. Also the completion may be symmetric or asymmetric. The latter case (the supporting leg is kept tucked while the other is extended) is called a "walkout" and it used to join the roll with other stunting movements which need a one - foot takeoff. An acrobatic variation Is the "backward extension roll" In which the arms are extended to a handstand in the middle part of the roiling movement. 8. Acrobatic Tumbling and Stunting The repertoire of acrobatic gymnastics is very large and open to inventions. (Indeed many difficult combinations are denoted by the name of their author). What follows Is a list of

Dance and Movement Notation 123 the basic movements, which are mainly characterized by the "struggle" against gravity. Handstand - In a handstand the body weight is supported by the hands. The handstand is reached from an upright position with raised arms, by bending the trunk forward, loading and bending one leg while throwing the other leg extended backward, and aiming the hands to two forward symmetric positions. Cartwheel - The cartwheel is essentially a combination of a handstand with a 160 degrees turn. From a starting posture, similar to that of a handstand, the hands are aimed at target positions which are aligned on the forward line which passes through the supporting foot (the closer target for the hand on the supporting side). The result is a rocking movement to a handstand with opened legs (which implies a 90 degrees turn). Finally, the same sequence of movements performed in the ascending phase is then repeated in a mirror fashion during the descending phase. The landing foot is the opposite one and the final posture is turned 180 degrees with respect to the initial one. For performing a sequence of cartwheels, it is necessary to perform a 180 degrees turn after each one. More complex variations of the same pattern can be performed with one hand or without hands. Roundoff - The roundoff has a similar geometry as the cartwheel but Is executed with more energy. (The appropriate initial procedure consists Indeed of a few running steps and a hop.) Differently from the cartwheel, the hands reach their targets simultaneously (although the closest hand is loaded first) and the body goes to a handstand with closed legs. In the descending phase (which is different from the ascending one) the legs are bent together and there is a vigorous push from the fingers and the shoulders. The landing in a demi-plie' guarantees a powerful loading, which may be very useful for subsequent stunts. In a backward roll or handspring, for example, it is convenient to land with the body off-balance backward but not when the next stunt Is a back somersault or another roundoff.(ln this case, since the roundoff implies a 180 degrees turn, it is necessary to perform a half turn between one roudoff and the next one.) Walkover, Back Walkover - Both these stunts are built on the "bridge" pose and on an inverted split. In the back walkover the bridge is reached bending the trunk backward. At the same time an extended leg is thrown up and the other follows to an inverted split, smoothly ending to an upright position. In the forward walkover the initial movement is similar to a handstand but then the legs move to an inverted split before going to the bridge posture and then to the final upright position. More complex than a walkover or back walkover, the aerial walkover. Aerial Walkover - is a special type of front somersault done in an arched position with a takeoff from one leg and a landing on the other leg.fTo get the right amount of initial energy, the performer starts with a hop.) Handspring, Headspring, Back Handspring - The handspring starts with a run and a

skip takeoff the same as that used for the roundoff: The hands, on the contrary, are aimed at two symmetric target positions in front. The body goes to a handstand and rebounds in the air (also pushed by the fingers and the shoulders) while the feet continue their circling arc over, to finally reach the floor in a straight or slightly arched body position. As a variation, the handspring may be finished with a walkout, which allows for example to repeat a series of stunts. The headspring should be started with a two - foot takeoff, then the hands and the fronthead are aimed to the floor (the head slightly in front) and the tumbler jumps or pushes from both feet Into a pike position until slightly offbalance backward. At this point the legs whip hard In a rotatory movement and the arms push hard off the floor, turning the body upright again. The same pattern can then be repeated indefinitely. The back handspring (also oalled flic flac) starts with a backward off balance posture and bent knees from where a rapid pattern is "exploded" (upward and backward swing of the arms, followed by a baokward extension of the head and a straightening of the legs) which pushes the body backward arched in the air. Slightly after the feet leave the ground, the hands touch down, with the body still arched backward. Finally, but not too soon, a hip flexion is started, the fingers and shoulders push off the body, and the feet are snapped down to the mat, ready for another flic flac. A series of back handsprings is usually started with a roundoff. Front Somersault, Back Somersault - In somersaults (also called flips) the body is turned over without an intermediate phase of support over the hands. While spinning In the air the body may be either tucked, or arched back, or bent in a pike position, or twisted15. Different types of arm movements are used, coordinated with the takeoff to provide height and spin (and twist, if necessary). Somersaults are often performed at the end of a sequence of other stunts, e.g. < roundoff, flic flac, back somersault> or < handspring, somersault >. " The tucked position is the position with the smallest moment ol Inertia and therefore the ona which allows the quickest spinning. To even Increase the spinning speed, e.g. in double aoumemHs, the tucked legs ara kept open and the head la bent Inside.

MOTOR KNOWLEDGE G. Marino, P. Morasso, E. Troiano and R. Zaccaria CONTENTS

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasco (Editors).

NEM: A LANGUAGE FOR THE REPRESENTATION OF MOTOR KNOWLEDGE G. Marino, P. Morasso, E. Troiano and R. Zaccaria The motor control problem is considered in the framework of knowledge representation, with special reference to motor knowledge. In the Al/Robotic world, a formal model for motor knowledge should fill up a gap between task planning and low level robot languages. Such a model should be able to "virtuallze" the robot and the interaction with the environment so that the planner could produce (and rely on) high level abstract actions characterized by high autonomy and skill. The chapter discusses some general aspects about actions, actors, and scenes that inspired the definition of the NEM language for the representation and animation of humanoids in a scene. It is meant to provide a software laboratory for experimenting with action schemes. 1. INTRODUCTION One of the main topics of Artificial Intelligence is presently knowledge representation. This means, in a broad sense, defining concepts and rules able to capture the eseential complexity of a given problem domain that escapes a direct approach to understanding. Thus, a knowledge representation (KR) approach emphasizes the goals and the functions of a system rather than its specific mechanisms, and in so doing it generalizes and abstracts our understanding of the system because it is likely that different mechanisms can be found that can implement the same function and goal. Abstraction and generality can be regarded at weak or strong aspects according to the point of view: weak for the neuroscientist, who can And only suggestive power in a KR model, but strong for a designer of skilled machines, who may rely on a KR model while developing a robot with anthropomorphic skills without being limited to the anthropomorphic "hardware". The richness of human skills cannot be understood if we do not adequately take into account the deep complementarity between a performing human (or, generally, an actor) and its surroundings. The "richness" of an actor's skill must match the "diversity" of Its surrounding. The actor/environment relationship may be characterized by a golden tetrahedron whose vertices are action, perception, cognition, and environment Each vertex of the tetrahedron is strongly linked to all the others by fundamental information and/or energy flows.

 Important theoretic contributions have been made by several authors on different aspects of the golden triangle, such as the concept of motor synergy [Bernstein 1967] or of motor schema [Plaget 1971, Schmidt 1975], the emphasis on the "ecological" nature of skills [Gibson 1979, Turvey and Kugler 1984], and the recognition of the linkage between perceptual structures and motor control [Arbib 1981]. Nevertheless, this kind of knowledge essentially remains at the conceptual level and does not provide the computational tools for performing, accordingly, "general" experiments, either in the real or in the simulated world. On the other hand, plenty of relevant methodological tools do exist, that deal with general spatio - temporal - logic topics at the base of action complexity: - computational geometry, 3D computer graphics, solid modeling; - geometric reasoning, task planning; - object oriented programming, message passing, actors; - logic programming, expert systems; - analogic computation, dynamics, compliance; - qualitative reasoning, naive physics. However, such tools are being developed In different and largely unrelated areas and one of the main problems that need to be faced is interfacing and Integrating them in the common framework of action. Taking all this Into account, this chapter summarizes some experimental findings about human movements, then discusses some general aspects about action, and finally describes a language, NEM, for the representation and animation of humanolds In a scene, which is meant to provide a software laboratory for experimenting with action. 2. A REVIEW OF EXPERIMENTAL FINDINGS The complexity of the motion of the human body has attracted the attention of first artists, and then scientists. Perhaps Marey [Marey 1873] and Muybrldge [Muybrldge 1901] can be chosen to mark the borderline from the artistic to the sclentlfio approach to human movement. However, immediately after the beginning of the scientific enquiry of human motion, it became apparent that the eidetic beauty of the human body had to be reduced to something more formal such as a stick figure. In mathematical terms, a stick figure can be represented as an ensemble of kinematic chains, I.e. as a structure of joints and links. The complexity of such a system, on the other hand, has directed the main stream of research on human movements to simpler subsystems of the body, characterized by "few" degrees of freedom. The overall human body modal has received attention mainly In computer graphics and computer animation [Potter and Wiilmert 1975, Kingsley et al. 1981, Herblson Evans 1983, Badler 1983] or In the fields of anthropometry and ergonomy, focusing on the range of permissible and/or comfortable movements that constrain the human body [Dooley 1982, Korein 1985].

NEM 129 A stereotyped spatio-temporal structure has been found to characterize simple rotations of different parts of the body, such as the eyes [Morasso et al. 1973], the head [Blzzi et al. 1976], the forearm [Polit and Bizzi 1983], i.e. rotations characterized by a smooth acceleration phase, followed by a smooth deceleration phase - a type of structure which has been shown to have a 'minimum-jerk" [Hollerbach and Flash 1982]. On the other hand, the joint level characterization of human movements is inappropriate for describing more complex movements, such as cursive script handwriting, as first suggested by Bernstein, when he formulated his principle of "equal simplicity" [Bernstein 1987]. This suggestion has been confirmed by subsequent investigations that showed the Invariant characteristics of hand movements performed in different parts of the workspace with different amplitudes and with different orientations [Viviani and Terzuolo 1980, Morasso 1981, Abend et al. 1982, Lacquanlti and Soechting 1982]. These movements also have bell shaped (minimum-jerk) velocity profiles and such a characterization implies a constraint on the shape and the speed of the trajectory. Specifically, the regular segmentation of the velocity profile and its coupling with the corresponding curvature profile suggest to associate each velocity peak with a primitive minimum-jerk motion and to decompose the overall trajectory accordingly as a chain of concurrent primitive motions [Morasso and Mussa Ivaldi 1982, Morasso 1983]. With regard to dynamics, the first thing to point out is that remarkable inertial changes, due for example to loads, have been reported to influence very little the spatlo - temporal characteristics of movements [Bizzi et al. 1978]. Furthermore, the viscous - elastic properties of the muscles have been shown to provide an analogic automatic feedback for the compensation of disturbances [Feld'man 1966, Blzzi et al. 1978], and their action oan be represented by means of a stiffness tensor [Mason 1981, Hogan 1982, Mussa Ivaldi et al. 1984). As a logical consequence of these findings, it has been suggested that each posture is the equilibrium point of the elastic forces due to the different muscles. The dynamics of the movements Is counteracted by shifting such a point along an appropriate trajectory [McKeon et al. 1984], The stiffness tensor may be expressed more intuitively In geometric terms by means of stiffness ellipses (In the plane, or stiffness ellipsoids, In the space) which specify the relation between vectors of displacement from an equilibrium point and the corresponding restoring force vectors, making the preferential directions of compliance explicit. The shape and orientation of the stiffness ellipses have been found to vary regularly in the workspace, therefore providing the possibility to formulate geometrically the compliance aspects of action. In general, the picture emerging from the experimental findings briefly reported in this review is that motor control problems may be divided into two grots categories: (a) energy driven processes that solve in an implicit/analogic way dynamic aspects of action, mostly relying on the mechanical properties of muscles and on the complexity of their geometric arrangement; (b) spatio-temporal processes, that express explicitly significant relations between an actor and the surrounding environment.

 3. ACTORS, ACTIONS, SCENES The goal of an action may be quite diverse, even discarding non motoric purposes. Let us consider some examples: (a) reaching or approaching someone or something according to a specific "trajectory" and with respect to a specific geometric framework; (b) hitting someone or something according to selective directional characteristics of the impact, i.e. according to a specific compliance: (c) catching and grasping someone or something by adapting the actor's self to the geometry and the physics of the grasped entity and so modifying the topology and the metrics of the actor's self; (d) pressing/pushing/pulling/twisting, i.e. applying a force field to someone or something according to a planned direction and intensity. By means of the four gross categories of general skills

outlined above (which correspond to generalized forms of positioning, impacting, pushing, and grasping), it is possible to compose an unbounded variety of actions characterized by a highly structured network of specifications. Geometric, kinematic, dynamic aspects are all tightly Interrelated for any skill. In general, however, it is convenient to distinguish, for an actor, two different levels of "knowledge" of the particular aspect of a skill: (a) a qualltative/symbolic/explicit level, and (b) an implicit/analogic/quantitative one. In the case of the dynamic aspects of a skill the distinction above corresponds to the so called "naive physics" [De Kleer 1975], in the former case, and In the latter case, to computations performed either by the mechanical properties of the muscles or by specialized signal processors1. Let us define an action as a purposeful change in the relation between an actor and its surroundings, which may consist of objects and/or other actors. Let us leave aside, for the moment, the action "purpose", which has to do with the non - geometric aspects of action planning: Let us consider the mutual "change" between the actor and his surroundings, which characterizes the specific motoric performance. It is important to realize that such a change is always multidimensional and highly redundant: Every significant geometric feature of the actor "changes" its geometric relation relative to any other geometric feature of the actor itself and of all other objects and actors in the scene. If we now associate with each feature, a local system of co-ordinates (a frame), the action can then be viewed as a stream of variations of some of the mutual relations between local co- 1 Specializad signal processors nave been delined lot "inverse kinematics' |Benatl el al. 1862. Lee 1983) and "inverse dynamics" |0nn et al. 1979, Benatl et al. 19B0, Luri et al. 1B80. Holiarbteh 1980, Paul 1981, Silver 1982|.

NEM 131 ordinate systems which are due to a stream of motor commands3. The description of the elements of a scene in terms of local CO - ordinate systems linked to features, needs obviously to be constrained in such a way that the actions do not violate the integrity of actors and of objects. Furthermore, it is convenient to group together frames referring to the same structure (an actor, an object, or a part thereof). As a consequence, a scene can be represented as a 'forest' of frames that changes over time. By visiting the forest it is possible to express the geometric relations between any two features of the scene. Some of these geometric relations can be interpreted as "motor' and some as "proprioceptive" whereas others can be interpreted as "visual' or, generally, "exteroceptive". Furthermore, It is worth noting that ensembles of frames can only represent, in a direct way, the "skeletal' structure of actors or objects; if smoother and more detailed representations are needed, it would be necessary to associate appropriate "shape formation" attributes with frames and frame ensembles. This points out the problem of Interfacing an action oriented system with both solid modeling concepts [Requicha 1980, Binford 1982] and techniques of path planning and obstacle avoidance [Lozano Perez 1982). It is important to suggest that the point of view about action discussed in this section is not limited to the so called "motor program concept" and It does not neglect the importance of "feedback" in motor control. On the contrary, both centrally planned and feedback driven movements fit in the same general framework, that stresses the geometrical complexity and abstract- ness of action. 4. NEM: A LANGUAGE FOR REPRESENTING ACTORS, SCENES, ACTIONS The NEM language3 has been designed to provide a procedural, non - hierarchical representation of motoric knowledge. Computational Models of Men (the New Men, from which the NEM name comes) are intended to: (a) define and model (potentially) moving entities ("objects" or "actors"); (b) describe the movement of an entity as a whole, or of groupe or parts of entities. Object/Actor representation is based on an atomic element called Qeometric Frame (QF), by which Qeometric Frame Structures (QFS's) can be built4. A QF can be associated with a body or a geometric feature of a body. Chains of QF's can be built for representing kinematic chains (arms, legs, etc). Chains can be assembled and branched, resulting in arbitrarily complex QFS's. 2 Motor commands are here delined. In a very broad sense, as desired variations ol geometric relations between features ol a scene. 3 The NEM programming environment consists ol a compiler (cnem), an executive (xnem), a loader (for complied libraries), and a debugger. The compiler (which includes a precompiler) is bated on the ihe C Preprocessor, the VACC compiler-compiler and other tools available under Unix for the DEC computer Vex 11-760. 4 "Qeometric Frames" neve obviously nothing in common with Mlnsky's frames; NEM Inherits thla term from analytic geometry and computer graphic*.

 From the topological point of view they are trees that provide a hierarchical representation for NEM objects: "finger", "hand", "arm", "humanoid" or "table_leg", "table_top", "table". Data bases of NEM objects can be built (see also the section on NIM, the NEM Input Module) and can be loaded at run time, setting up the "environment" for a NEM action program. Moving objects are called Actors. An Actor's movement Is described by means of scripts called motions Motions are abstract, non - hierarchical procedural descriptions of movement for single objects, perls or groups of them, or generalizations of object families5. A motion can adapt itself to a particular object in a family. For example, a script "anlmal_walk" could fit, when activated, different actors like a "spider" or a "dog". Motions are built by composition, or specialization of other motions. Motions can express geometric reasoning, both static ("Is the book over the chair?") and dynamic ("may John reach the book without walking?"). Finally, motions allow expressing common sense motoric knowledge (what we may call "common skill knowledge"). Though NEM is devised to express only kinematic entitles, naive physics concepts such as "gravity", "equilibrium", "collision", "pushing" and so on can be easily defined in terms of general virtual motions. A proper algebra called Frame Algebra Notation (FAN) has been defined: It Is a formal notation that is embedded in the language and allows to manipulate QFS's and to support Geometric Reasoning. FAN is one of the three components of NEM: The other two are Motion Conatnictors and Motor Primitives. The former are the "glue" by which motions are build and held together - rules for movement composition and motion definition, methods for writing "templates" of actions, and so on. The latter are the atomic (low level, built-in) motor primitives and their semantics. The detailed presentation of NEM that we give in the following sections is not intended to be a user manual but only to outline the significant features of the language as a tool for the study of movement. Although NEM is not a general purpose programming language, it has most of the standard features and constructs of general purpose programming languages, particularly the C language. Since NEM is an experimental project, great attention is being paid at making it a flexible experimental tool, in which new primitives can be added and old ones can be modified. In this respect, the Unix-C environment proved to be quite appropriate. When describing the nature of a programming language, one can either outline the syntax and semantics of its phrases or emphasize the hierarchical levels of abstraction that the language can manage. The two ways will be mixed in the following sections. Standard programming languages are suited for a narrow set of levels. This Is not the case of the "natural" language, whose characteristic is the ability of mixing very abstract ooncepts and punctual, detailed statements, particularly when it is used for describing sansory - motor actions [Adorni et al. 1984]. For this reeson, NEM is designed as a multi -level language: Its primitives range from low levels (e.g. frame manipulation) to high levels (gsnerlc action templates definition, dynamic block instancing, ...). 5 In NEM the distinction between "tamilles" and "objects' is internal: a 'family* is an objact definition in parametric terms.

NEM 133 4.1. Data Types and Frame Algebra Notation Four data types are defined in NEM: frames triples scalars 6 events In NEM a frame plays at least three complementary roles: 1. Frames can be regarded as Cartesian systems of reference linked to physical objects and/or to their elements. (The information stored by the frames can be interpreted as object position/orientation.) 2. Fremes can be used to express the geometric relation between two objects, i.e. the relative rotation and translation. 3. Frames can express object movements because any movement is a combination of a translation and a rotation. The abstract geometric concept of frame is represented computationally by means of 4x4 homogeneous matrices [Foley and Van Dam 1983]: . x R .y H « .2 00 0.' 1 Homogeneous matrices are composed of a rotational part (a 3x3 matrix R, that stores the direction cosines of the co-ordinate axes) and a translational part (the 3 - dimensional vector [x,y,z|, that identifies the origin of the frame)7. Frame Declaration Frames can be defined by means of the FRAME declaration, which also allows to express the relation between frames by means of the & operator. For example FRAME A ®A FRAME F declares two frames (A and F) and states that F expresses a rotation/translation relative to A: A is the ancestor frame of the son frame F. The default ancestor (the father of all) is ENV, the "environment". An ancestor can have many sons and chains of frames can be cascaded, generating complex frame structures. For example, a solid object (a pyramid) can be defined by the following frame structure: ' Scalers are Intended (o support the simple data types as In most programming languages (namely Integer and lloat). 7 Mathematical details concerning homogeneous matrices are given In an appendix of the lirst chapter. The NEM Internal representation ot homogeneous matrices uses 3x4 arrays: the last row (0,0,0,1) Is Implied by the semantics ot frame operations.

134 G. Marino et al. ® ENV FRAME PYRAMID ® PYRAMID FRAME VERTEX_1 VERTEX_a VERTEX_3 where ENV is the "environment" frame, PYRAMID is the frame of the father-vertex and the other three frames correspond to the three son - vertices. If PYRAMID la moved then all the other

vertices are moved in a rigid way. The composition of complex QFS's is aided by the availability of operators for forming chains of frames and for duplicating them. For example, the QFS of a humanoid skeleton may be declared as follows: ® ENV FRAME HUMANOID ® HUMANOID LIMB PELVIS TORSO NECK ® PELVIS LIMB 'L' ILIUM HIP KNEE ANKLE FOOT ® PELVIS LIMB 'R' ILIUM HIP KNEE ANKLE FOOT ® TORSO LIMB 'L' SCAPULA SHOULDER ELBOW WRIST HAND ® TORSO LIMB 'R' SCAPULA SHOULDER ELBOW WRIST HAND ® NECK FRAME HEAD ® HEAD FRAME MOUTH NOSE FOREHEAD ® HEAD FRAME 'R' EYE EAR (Figure 1 shows the image of the humanoid QFS generated by the graphic interface. The Interface draws the segments which connect the origin of any "lighted" frame to the origin of its ancestor frame.) Here "LIMB" defines a chain of frames (e.g. pelvis is the ancestor of ilium, which is the ancestor of hip, which is the ancestor of knee etc.) and the prefixes ("R", "L") denote distinct elements with the same structure (e.g. the frame associated with the "right knee" is named R'KNEE). A QFS can be regarded as rigid as well as articulated. In a rigid QFS (e.g. the pyramid QFS) one should move only the father frame whereas in an articulated QFS (e.g. the humanoid QFS) several frames can be moved, specifically, those corresponding to joints. Representing, generating, and duplicating QFS's is greatly aided by the use of macro definitions. For example DEFINE HUMANOIDO LIMB PELVIS TORSO NECK Is a template QFS that can be instanced in multiple and different copies: FRAME ADAM EVE ® ADAM HUMANOID() ® EVE HUMANOID() Macro definitions are a way for implementing data abstraction; their use is Illustrated more extensively in the following. Initialization and Decomposition of Frames The triple is the atomic element from which it is possible to build a frame. The triple is a 3- dlmenslonal array which can store, among other things, a translation vector

NBM 135 Figure 1. Humanoid frame geometric structure: the stick figure on the left (the standard graphic output) displays the vectors which join the origin of a (lighted) frame to the origin of Itt ancestor; the center figure displays only the frames ol the humanoid, and the right figure shows the frame structure together with the stick figure. a sat of direction cosines (i.e. a unit-vector or an axis) a set of Euler parameters a sat of Rodrlgues parameters8. Triples are defined by TRIPLE declarations and can be assigned sets of scalar values as In the following example: TRIPLE X_AXIS,Y_AXIS,Z_AXIS,ORIQIN X_AXIS-

(0,0,1) Y_AXIS-(0,1,0) Z AXIS-(1,0,0) ORIGIN-(1,2,3) By means of triples, frames can be initialized as follows: ® ENV FRAME TRANSLATED_F [TRANSLATED_Fl -[X_AXIS,Y_AXIS,Z_AXIS,ORIQIN] e The above defines a frame oriented in the same way as the reference frame and displaced by the vector (1 ,a,3). The complementary operation, as regards initialization/synthesis of frames, is frame decomposition/analysis. Several operators are defined in NEM to perform this. For example, the operator pos(F) extracts from a frame F the vector which joins the origin of the ancestor frame to the origin of F. Other operators 8 The direction cosines are the scalar products of a unit-vector with the three co-ordinate unit-vectors, i.e. they are the Cartesian components ol a unit - vector. Euler parameters are three angles which Identity a rotation matrix. Rodrlgues parameters Identity the seme matrix in a different way (see the appendix ol chapter 1). • In a frame assignment it is uselul to use the shorthend notation ' -' to denote zero rotation or zero translation. For example (TRANSLATED_F]-1-ORIGIN) denotes a frame that is oriented as the environment frame and le shifted with regard to It.

 xdlr(F) ydlr(F) zdir(F) extract the direction cosines of the x, y, z axes, respectively, and eul(F) rod(F) extract the Euler and the Rodrigues parameters, respectively. Finally, we may mention operators which transform a triple in a triple, such as xlue(triple) ylue(triple) zlue(triple) that transform a triple of Euler angles into the corresponding x - axls.y - axis, and z-axis, respectively, or xdor(triple) ydor(trlple) zdor(triple) that perform the same operations on a triple of Rodrigues parameters. Operations on Triples and Frames In addition to the primitives of movement, that are examined In the following, NEM has many operators, on triples and frames, that are particularly useful for expressing geometric reasoning. Specifically, the standard vector algebra operators are available: vector sum + vector difference vector product « scalar product With them It is possible to express complex geometric entitles. For example, in a "placing" task it may be necessary to find the direction perpendicular to the planar face of an object when we know three non colllnear points of the face: P1 P2P3. FAN allows to express this relation in the natural geometric way, i.e. by means of a vector product: TRIPLE DIRECTION DIRECTION - (P1-P2)x(P1-P3) Another significant problem is to express geometric relations between two objects (identified by two corresponding frames). Let us suppose, for example, that we are interested in the distance between two objects. If the two frames (F1 and

F2) are direct relatives (one is the ancestor of the other), then the calculation of the distance is simply the following: ®F1 FRAME F2 DISTANCE - NORM(POS(F2)) 10 Otherwise, if the two frames are not relatives, their relation can be made explicit by means of the notation F1 ® F2 as in the case NORM normalizes a vector to unit - length.

NEM 137 F - ADAM'MOUTH ® EVE'R'HAND which stores in the frame F the geometric relation existing, at a certain instant of the simulation, between Adam's mouth and Eve's right hand. Naturally, on this frame it is possible to apply all the FAN operators. Another operator on frames is the frame product (denoted by "*") that corresponds to chaining or cascading the corresponding transformations. For example, the notation FT Fa Is a frame that is equivalent to the composition (one after the other) of the two transformations F1 and Fa. (Note that the order is significant, because matrix multiplication is not commutative.) Therefore, the following calculation (F1 ® Fa)* Fa gives back F1. What is important is that complex geometric concepts can be formulated in FAN and so may become primitive geometric entities in the context of a task. This feature of FAN is enhanced by macro definitions. For example, a special function which computes the distance between two frames can be defined as DEFINE DISTANCE(F1,F2)-NORM(POS(F1 ® Fa)) and the abstract relation "above" can be defined by the following formalism: DEFINE ABOVE(F1,Fa)-Z(Fl ® ENV) > Z(Fa ® ENV) 11. Another convenient feature of FAN is the pathname. A pathname takes into account that QFS's are structurally trees and this allows to reference a frame not only by Its name but also by its location in the QFS, i.e. its pathname. For example, ADAMI2I2M refers to the < first of <the second of <the second descendant of <the frame ADAM> > > > and is a synonym (in this particular case) of the frame <ADAM'R'SHOULDER>. Pathnames and related functions12 allow motion scripts to fit also structures, not only names. Events Events are FAN formulae that express some specific aspect of an action: e.g. "heel strike" in walking, "target too far" in reaching, "hand aligned" in manipulation, etc. Events may be used in different ways. For example, at a coarse level of representation they may be generated as messages from one ector to another, from one skill to another with the purpose of some kind of cooperation. At a lower level of representation, e.g. inside a skill, events may be regarded as " Z( ) is a lunction of a Irame that gives back the z-component of the displacement vector. '>' is the relational operator 'greater then'. Other standard relational operators are available ><>.<- •• I- as well as

Boolean logic operators AND OR NOT. 12 For example, a usalul pathname function is IS_DEF, which tests whether a certain element of a QFS exists or not

138 G. Marino et al. guards that constrain the execution of some pattern to specific conditions. But that Is not all: Other meanings of "event" may be conceived in the framework of specific tasks. In any case, the concept of event implies to focus the attention on some perceptual pattern which is tested over time as long as required. Therefore, events are perceptual as well as motor entities. Event specification is a sort of perceptual programming that complements the motor programming aspect of the NEM environment. The meeting point between perceptual and motor programming is centered on FAN structures, i.e. geometric reasoning. 4.2. MotionB Motor patterns are built at three levels of complexities: 1. primitive motions 2. composite motions 3. motion blocks where the first level corresponds to the basic geometric methods of movement shaping, the second level represents the different ways of constructing concurrent motor sequences (possibly sensitive to environment information) and the third level is suitable to represent a motor skill. It is Important to point out that several motions can concurrently operate on the same QFS. This is one of the most powerful mechanisms for the smooth generation of motor patterns. Primitive Motor Operators Atomic elements for motion construction are primitive motor operators (PMO) that are divided Into two classes: joint primitives, and limb primitives. A joint primitive rotates a frame with respect to its ancestor along different axes: x - axis of the ancestor BEND y-axis of the ancestor FOLD z - axis of the son ROTATE node axis STRETCH arbitrary axis TURN screw axis SCREW BEND, FOLD, ROTATE, and STRETCH perform rotations in the flavor of the Euler angles. The "node axia" in particular Is the axis perpendicular to the plane formed by the z-axes of the two frames. It is directed In suoh a way that the clockwise rotation from the son to the ancestor is less than 180 degrees. TURN performs a rotation in the flavor of the Rodrigues parameters. SCREW does not move the specified frame but it moves its ancestor - ancestor with a rotation around an axis which joins the frame to its ancestor - ancestor13. The rotation angle, as well as

NEM 139 the frame name, is specified as a parameter ('turn" requires to specify also an axis). Limb primitives involve the co-ordinated rotation of more joints and involve spatial relations, not joint relations. All of them operate on a three-link chain, i.e. the named frame (considered as the end-effector of the kinematic chain), its ancestor, and its ancestor-ancestor. Specifically, MOVE drives the end effector toward a target frame; EVOM performs the complementary operation (an inverse move) which keeps the end effector fixed and moves the base of the limb; FIX specifies that the end effector has to keep its current relation with respect to a target frame, until an UNFIX command is issued. These primitives involve Inverse kinematics, a standard feature of NEM14. We ooukj say that primitives at the limb level are a way for localizing motor information where really required. Thinking of a humanoid, MOVE could define, for example, the motion of a hand, without caring of how the arm follows It; FIX could assure that a hand remains still whatever happens to the rest of the body; EVOM could specify how the hip of a dancer moves while the corresponding foot 'moves* along the floor. PMO's execute in a single time slot, i.e. in a single cycle of the NEM simulation process. In fact, they are the source of time flow in NEM because all the other computations are considered to require zero-time. NEM simulates the advancement of time in a way conceptually similar to simulatbn languages like SIMULA or QPSS: at any time instant (or time-slot) a number of processes (actors or motions in NEM terminology) may be simultaneously active and their operations are performed in turn (pseudo - parallelism). A time slot is assumed to have constant duration. If we wish to perform a movement (even a PMO) with a desired velocity profile (with a duration, therefore, that extends over many time slots) we must break down the movement into a sequence of movements. The amplitude of any of them is proportional to the desired instantaneous speed. Composite Motor Operators Composite Motions are defined by combining primitive and/or composite motions by means of sequencing operators which implement three different types of execution: 1. Sequential Execution: ACTION_1 ; ACTION_2 ; ACTION_3 ; ... The execution of "ACTION_2" is initiated only after "ACTION_1" Is terminated, and so on, 13 The sorew primitive represents a typical movement ot the arm. e.g. when turning a screwdriver The movement ol the shoulder is constrained to keap the same position ol the screwdriver. 14 The algorithm ol inverse kinematics implemented in NEM is based on the paper |Benatl et al. 1962], which is reviewed In the first chapter.

 where "ACTION" stands for a PMO or a composite motion or a motion block. The global duration is the sum of the durations. 2.

Parallel Execution: ACTION_1 & ACTION_2 & ACTION_3 & ... "ACTI0N_1", "ACTION_2", are started simultaneously. The global duration is equal to the longest individual duration. 3. Guarded Execution. three different paradigms are implemented 1. WHILE condition DO action 2. WAIT event THEN action 3. PERFORM action EXCEPT condition_1 THEN recovery_act_1 EXCEPT condition_2 THEN recovery_act_2 ... In the first case, the execution is started immediately and It is continued as long as a "condition" is met. In the second case, the execution is started after a certain "event". In the third case, the execution of the action is continued until one of the guards becomes true and then a recovery action is started. Motion Blocks A motion block15 Is a program unit with a name, parameters, a declaration of Internal and external variables, and a body which specifies a motion pattern. Let us consider, as an elementary example, a motion block that generates an instance of a PYRAMID and rotates it until an external event stops the rotation. The pyramid then "opens up" by moving the three vertices simultaneously16: MOTION FUNNY PYRAMID(AXIS) TRIPLE AXIS EXTERN EVENT STOP ® ENV FRAME PYRAMID() INIT PYRAMID(< parameters >); perTorm turn pyramid around axis of 2 deq except stop THEN { T - ACTUALTIME; WHILE (ACTUALTIME < fT + 10)) DO BEND VERTEX_1 OF 1 DEQ & BEND VERTEX_2 OF 2 DEQ ft BEND VERTEX 3 OF 3 DEQ }}} 15 A NEM block is a "chunk of code" enclosed in f) brackets; it ia preceded by a header In which local and external variables are declared. Blocks may be used to define QFS's, Motions, Adore, aa well aa standard malhematloal functions. Blocks are program units that, similarly to the C language, are all at the same level.

NEM 141 Frames and variables may be local or global (common to other motions). We may add, for computer hackers, a few details about NEM semantics. Global data may be bound to parameters of the motion call, or to other global variables and frames. Since the semantics of a motion is, in general, context-dependent, the binding of global variables and frames Is necessarily dynamic. In fact, when a motion is activated to provide actual movement of an actor, binding of variables has the meaning of interaction with other actors and the environment in the actual context. Motions are similar to LISP functions, except for two facts: i) motions can be activated concurrently, and ii) composition of motions allows non - hierarchical programming. Access to data or to motion names Is controlled by simple rules. Motion names are global, i.e., they can be accessed by every other motion; however, libraries17 can be built to form contexts for related motions and to allow multiple definitions. Global data are static, i.e. they live forever, starting from the first time a declaration is found during execution. Local data cannot be accessed outside the motion that declares them, and are dynamic (or 'automatic* in the C Language terminology), i.e. they disappear as the motion returns. However, local frames may be linked to a global ancestor, in this way, dynamic GFS's may be accessible to any other motion. The motion composition mechanisms allow to overlap in time different motion patterns that refer to the same GFS, therefore providing an unlimited capability of trajectory formation. In walking, for example, the complex and variable swinging movement of the leg in the swing phase of gait can be represented as the concurrent activation of a motion, that aims the swinging foot toward its target foot print, and another motion which tends to shorten the hip-ankle distance (with an amplitude which is a function of the terrain and the presence of obstacles). Another trivial example is a motion in which the patterns generated by FUNNY_PYRAMID are concurrently activated with other (funny) motions: MOTION MORE FUNNY PYRAMID(...) { FUNNY PYRAMID(...) & FLY_PYRAMID(...) & FLIP_PYRAMID(...) & } where FLY_PYRAMID could be a motion which flies PYRAMID along some spatial path, FLIP_PYRAMID could be a motion which turns PYRAMID upside down and so on. The composition paradigm allows to express naturally a non - hierarchical, distributed approach to motor control [Hinton 1984] which is conceptually akin to the object-oriented programming style [Weinreb and Moon 1960, Goldberg 1983]. In such a framework, "central programming" and "peripheral feedback" are fundamentally and unavoidably Intermingled. Guarded executions allow ectors to synchronize through events, or to communicate with each other through message passing. " In this example, two macros are used: PYRAMID() and INIT_PYRAMID<). They define a suitable GFS and initialize it, respectively. " Libraries ot motions and ol macro definitions are related to the ADA concept of package.

 4.3. Generic Template Definition & Dynamic Block Instancing Macros were already introduced as a tool for abstraction. In this section macro definitions are used to write generic templates™ of actions. When representing an action, it is significant to separate (a) a level at which we define in a generic way the structure of different objects and actors (as

QFS's) and the structure of different motion skills (as motion blocks), and (b) a level at which individual instances of such templates are produced and a specific "happening" is triggered. Moreover, there are problems of locality and globality of information that can be better explained by means of an example. In particular, let us assume that we wish to represent the action of a ballet. There is a scene and there are some dancers, say three (John, Mary, and Alex). We need a formalism to describe and generate the scene and each of the dancers, as Individual QFS's. We need also to define the "script" of each dancer and the overall ballet script. The scene QFS must be global, i.e. available to all the actors, and static, I.e. permanent during the ballet. The dancers QFS's must the global but dynamic, i.e. a QFS must be Instanced when the corresponding dancer enters the scene and it must be deleted when he exits. Moreover, if different dancers must perform the same generic acript, it is desirable to avoid writing it several times while taking into account the minor individual differences and the "titularity" of the script. A useful concept, to taokle the complexity of action representation, is that of template. A template of a generic dancer is defined by means of a maoro definition DEFINE DANCER() LIMB PELVIS TORSO NECK and motion templates can be defined in a similar way DEFINE DANCES WALTZ(NAME) MOTION DANClS_WALTZ'NAME < declarations > { < motion body > } The mechanism of template expansion allows to generate several symbolic copies of the same template and this is equivalent to re-write the corpus of a macro taking into account the macro parameters. In the case of DANCER, for example, the parameters may specify some specific structural feature of the individual dancer (e.g. tall, fat, ...) and in the case of DANCES_WALTZ the parameter NAME has the function of generating the (individual name) of the motion (e.g. DANCES_WALTZ'JOHN, DANCES_WALTZ'MARY). 18 The concept of a generic template has been introduced by the ADA language (where It is called simply "generic") as a fundamental programming concept. Its purpose Is the possibility of writing not only programs, but also very abstract definitions of programs (or templates) Iran which to derive actual particular versions: lor Instance, a generic stack pro- oHurt could be written lo derive, in turn, a "stack of integer", a "stack of float", or a "stack of T" (where T la some type defined elsewhere).

NEM 143 The ballet action will consist of different motion blocks that will involve the (three) dancers in various combinations. Suppose that we can divide the ballet in two parts: in the first part John and Mary perform together (without Alex) and in the second part Alex performs alone. The ballet action19 can be represented as follows: ACTION BALLET() FRAME JOHN ALEX MARY {'{ ENTERS'JOHN & ENTERS'MARY & {RUNS'JOHN; JUMPS'JOHN; DANCES_WALTZ'JOHN; ...} & {RUNS'MARY; TURNS'MARY; DANCES_WALTZ'MARY; ...} } { ENTERS'ALEX & {ROLLS'ALEX; JUMPS'ALEX; ...} } The first part of the ballet is represented ss a block: the two performing dancers are dynamically generated and at the same time their "scripts" (which are sequences of motions) are activated. When the performance is over the two actors disappear because the block terminates and the block corresponding to the second part becomes active: Alex is generated and its "script" becomes active. The mechanism for the dynamic generation of QFS's which correspond to the different actors uses the concept of local or dynamic frame in a peculiar way. The main ballet action declares indeed JOHN, ALEX, and MARY as global frames, which are visible to everybody. On the other hand, the different instances of the motion ENTERS have the specific function of "supporting" the existence of the corresponding dancer. This may be done by the following template definition: DEFINE ENTERS(NAME) MOTION ENTERS'NAME ® NAME LOCAL FRAME DANCER() { < motion body > } The template must be instanced as many times as needed (in our case three) in such a way that it may be visible by BALLET. In this way, for example, ENTERS'JOHN" generates the "body" of JOHN (i.e. a QFS linked to the frame JOHN) that disappears as soon as ENTERS'JOHN is terminated (because ENTERS generates a local/dynamic QFS). However, this QFS will be visible to everybody because it is linked to a global frame (JOHN). The sequencing/synchronization of the different movements of the three actors is expressed in two ways, at two different levels. On one level, the ballet action itself makes explicit the coarse structure of the ballet: for example it says that John dances together with Mary and that he will be engaged in a certain sequence of "motions". The finer details, on the other hand, are written Inside the motions and are executed, for example, by means of "message passing". 'e "ACTION" is a synonym of "MOTION". This emphasizes the double nature of a motion or action block. In some cases it corresponds to the concept 'structured set ol movements ot an actor' and in other cases it depicts a collective happening.

 5. EXAMPLES The following simple examples of

execution of NEM scripts are aimed to point out the most important characteristics of the NEM language in the framework of Knowledge Representation. In particular: ability of expressing abstract motoric concepts, naive physics and common skill reasoning, and of acting as an analog model. Handwriting Motion Figure 2 shows an example in which a handwriting activity (the word "lenny") is modeled by a concurrent stream of atomic movements called "strokes", according to the computational model described in the first chapter of this part. "Strokes" are generalized MOVE'S characterized by direction, length, roundness and generated with a smooth velocity profile. It is important to point out the generality of this representation since the same motion remains valid for different sizes of the letters, different orientations, and different locations. Finally, It can be combined with other motions to vary the shape of scripts, and may apply, with different degrees of accuracy, to other QFS's than the right arm (e.g. a leg, the mouth), provided that they have a "similar1' frame structure. Walking Motion Walking (figure 3) is a combination of both joint and space level primitives using sequential, parallel, and guarded constructs. The motion is defined at several levels. The highest level is a process that moves the center of gravity of the humanoid along a given trajectory and with a given speed; lowest level actions (arms, legs, etc.) are autonomous and synohronlze among themselves and the highest level implicitly by message passing. The walking skill Implements, In part, the formalization described in the second chapter of this part. "Walk" is general in the sense that speed, step length, and trajectory, fitting to the particular humanoid, are parameterized or implicitly solved; furthermore, new specialized versions of It can be written by instancing "walk" with specific details. The role played by "events" In the NEM walking program is an example of the fundamental Interaction between central control and peripheral feedback which can be expressed in the language. Oving Motion Figure 4 shows a diving humanoid. It is worth noting the action of gravity, which determines, concurrently with everything else, the motion of the center of gravity. This Is an example of physical laws implicitly embedded Into the motoric knowledge. Though NEM does not Include dynamics, dynamic aspects can be modeled by kinematic processes. In the actual case, a gravity process Is active in concurrence with other motion processes and modifies the planned action. However, such "environmental" processes can be tested by "control" actors In order to tune action parameters.

NEM 145 Figure 2. Handwriting NEM Program Figure 3. Walking NEM Program

146 G. Marino et al. Figure 4. Diving NEM Program

NEM 147 Picking Up Motion A particular "pick up action" is shown in figure 5. It is one of the infinite actions that this motion can perform. PICK_UP(HUMANOID, OBJECT) embodies the knowledge about "generalized reaching", i.e. the skill of reaching for an object within the grasp of the actor. The motor skill will generate only movements of the arm if the object is close enough, or co-ordinated movements of the body, as in the case shown in the figure. If the object is even further away an additional phase of walking is required. The smooth chaining of gross motor skills, like walking and reaching, is one of the most interesting aspects of human motor performance: There is a lack of experimental data and we are just beginning to model the problem. For example, all the experimental activity on locomotion Is focused on the steady state patterns without paying any attention at the transition phases (initiation, termination, change of direction or rhythm etc.). Besides the high level of abstraction of the motion, another characteristic is the activity of an equilibrium process which forces the center of gravity of the humanoid to fall onto the geometric center of its feet by triggering proper "reflex" corrective actions of the legs and of the arms. This principle is an attempt to formalize a high level of "common skill" knowledge and enhances the capabilities of supporting a formal analog model of the mechanical world. 'Sitting Down' Motion The three paradigms discussed above: parameterization (abstract motions), gravity (physical laws), equilibrium (implicit or "unconscious" motions), are present even in the simplest everyday activities. Figure 6 shows the SIT_DOWN(HUMANOID, CHAIR) motion. The motion makes a humanoid sit on a chair of any height between floor level and knee level. During the first phase of the motion, the humanoid bends itself in a quasi-static

148 G. Marino et al. equilibrium condition; in the second phase, the humanoid releases the equilibrium process and falls (softly)) on the seat. The two phases are partially overlapped and implicitly synchronized with the actual environment, in this case, the height of the chair. Obstacle Avoidance In reaching as well as in locomotion, the primary goal of following a path in

space is always associated with an implicit goal: avoiding obstacles if they occur along the planned path. Although this problem may be formulated as explicit planning obstacle - free paths, In many cases we think that a simpler mechanism may apply (as it is discussed also in the chapter on planning). The mechanism consists of an Obstacle Repulsive Actor, I.e. a NEM motion or demon that senses the distance from (or the contact with) obstacles and generates a repulsive motion which acts concurrently with the plan (e.g. a straight path). As shown in figure 7 (the planar case of a punctual actor is considered, for simplicity) the same plan, with different obstacle configurations, may originate quite different paths. This kind of algorithm does not guarantee a perfect performance in all situations but it works satisfactorily in many cases, even with approximate Information about the obstacles. Moreover, such Information does not need to be global, as in direct planning. Only a rough perceptual process is required that interacts with the main motor program. The latter, however, may take into account the structure of the environment and the presence of obstacles but it does not need to be very accurate if it can rely on a run-time mechanism such as that described above. A synergistic co-operation between a motor plan and a perceptually driven obstacle repulsive actor seems to be an Interesting approach for problems of manipulation that require to reach objects in a very cluttered environment. Figure 6. "Sitting Down" NEM Program

NEM 149 Figure 7. Obstacle Avoidance 6. THE NEM GRAPHIC INTERFACE Figures 1 through 7 were generated by using the standard NEM graphic interface. This allows to define the position of the observer with regard to the environment and to visualize, using perspective projections, the vectors which join one frame to Its ancestor. It is possible to select which vectors are visible and which are not. In this way a wireframe representation of selected details of the QFS's can be obtained. A more sophisticated graphic output can be obtained interfacing NEM QFS's with solid modeling packages such as PADL - 2. Figure 8 was generated In this way. The interface is particularly natural because PADL-2 allows to build complex three-dimensional shapes and to identify them with a Cartesian reference and a name (see also the chapter on shapa and action). In this way, it is possible to define solids for the different body segments (head, torso, neck, humerus, forearm, hand, etc.) which are then "glued" onto the corresponding frames of the NEM QFS's. Such a PADL-man is articulated because the underlying NEM-man Is articulated. The only notable drawback of this interface is the requirement of significant computer resources (memory and time). 7. NIM: THE NEM INPUT MODULE NIM is intended to build up a data base of human body postures from real data in order to use them as "starting postures" of NEM action programs or, more In general, as "reference postures". NIM uses a set of images taken from different viewpoints by means of photographic or television cameras (see figure 9). The body is viewed Inside a calibration object (a wireframe cube in the figure). NIM is interactive (the user provides data points by means of a digitizing tablet) and proceeds in two phases. In the first one it estimates, from the set of images, a three- 3p^

150 G. Marino et al. Figure 8. The PADL - man dimensional stick-figure model of the body, i.e. the 3-D Cartesian co-ordinates of all the "significant" points of the body. In the second one, NIM transforms the stick figure model into the NEM frame model. In the initial phase, NIM uses "ray firing" techniques (commonly used In computer graphics applications). Synthetic rays, i.e. the equations of straight Unas, are used for two purposes: (I) to estimate, for each image, the three-dimensional location of the "center of projection" (COP), and (II) to calculate, from many images, the three-dimensional location of a selected point (e.g. a knee) simultaneously visible in at least two images. The basic fact which Is being exploited is that, according to optic geometry, an Image is formed by Infinite rays of light, all of them converging In the COP. Calculation of the COP The purpose of the calibration object is to provide a known geometric framework common to all the viewpoints, against whloh the "ray firing" can be formulated, If we associate a Cartesian

k ; 1 i V 4 ~^ A i Figure 9. The multiple images used by NIM CI

152 G. Marino et at. reference system with the calibration object, then we have as a priori, the 3-D co-ordinates of all its vertices. For estimating the location of the COP for a given image we can proceed as follows: 1. We single out the background vertices which are contained inside a visible foreground face and we digitize their planar co-ordinates as well as the co-ordinates of the foreground vertices (in arbitrary units, it only matters that all the measurements are "registered"). 2. For each background vertex, we subdivide recursively the foreground face containing its image until It is small enough to be considered a pixel. (This procedure yields also the 3-D co-ordinates of the point In which

the ray intersects the face.) 3. The equation of the ray is readily written. (It is the straight line which joins the a priori known background vertex and the computed point of the foreground face.) 4. The intersection of two or more rays is estimated by means of the least squares method: This is the COP. Calculation of the 3-D Co-ordinates of the Stick Figure A very similar computational procedure can be applied for estimating the location of the points of the stick figure after the locations of all the COP s have been computed. In particular, the equation of the ray which corresponds to a given point on a given Image can be written if we proceed as follows: 1. For any image in which a selected stick figure point is visible, we consider the ray from the COP to the point and we compute its intersection with the foreground face of the calibration object which contains the image of the point using the same recursive subdivision procedure mentioned above. 2. We write the equation of the corresponding ray (given the Intersection point and the COP). 3. We estimate the 3 - D location of the point of the stick figure by computing the least square intersection among all the rays. Calculation of the Humanoid QFS The second phase of NIM must construct the humanoid QF8 from the collection of 3-D coordinates of all the points of the stick figure. Specifically, we must take into account the structure of the homogeneous matrices that are used to store the frames and the conventions about joints, links, and co-ordinate axes. For example, the ELBOW frame of the NEM humanoid Is characterized by the fact that the z-axis is directed as the forearm, and the x-axis is perpendicular to the plane (wrist, elbow, shoulder). The initialization of ELBOW requires that we know the numerical values of the ancestor frame SHOULDER, and the 3D vectors of wrist (W), elbow (E), shoulder (S); the computation proceeds then as follows (we express it In the NEM formalism): TRIPLE E,W,S,X,Y,Z,V

NEM 153 FRAME F V«(E-S) Z«(W-E)/NORM(W-E) X - (W - E)x(E - S)/NORM((W - E)x(E - S)) Y-ZxX [Fl-[X,Y,Z,Vl ELBOW -F ® SHOULDER The last operation is necessary because E,W,S are computed with respect to the environment (so F is the frame ELBOW ® ENV) whereas the semantics of the humanoid QFS requires ELBOW to be referred to SHOULDER. 8. NOM: THE NEM OUTPUT MODULE During the execution of a NEM action program, it is possible to store a complete history, I.e. the image of a selected set of the QFS's that represent the environment of an action. Obviously, such an history is hardly a significant notation in itself. However, It is possible to extract from It parameters, descriptors etc. and present them according to some eidetlc format. For example, the trajectory of the center of gravity is not explicitly stored but it can be computed from the history and it can be represented in 2-D or 3-D form. Similarly, the Joint rotation angles of the Eshkol-Wachman notation system could be computed from the history and their time course could be visualized on a corresponding large staff, together with a subdivision of the time axis into (numbered) measures. Therefore, Norn is an open part of NEM which can be specialized according to the needs. The part which should be common to all the future developments is a formalism for extracting descriptors, parameters etc. from the history. Such a formalism is the same geometric formalism of NEM, used the other way around, i.e. as a tool of analysis instead of synthesis. Norn is presently in an advanced phase of development. 9. LANGUAGES FOR MODELING OBJECTS AND ACTIONS The NEM language has been designed as an attempt to face the lack of formal models for representing human movement, in the general framework of Knowledge Representation. However, NEM takes advantage of many concepts which have emerged In different areas of non conventional computer programming, particularly in simulation of processes, In solid modeling, in computer animation, in robot programming, in artificial intelligence. We might classify the common issues in two main topics: Object/World Modeling Action Modeling

 Object Modeling When modeling scenes, objects, humanoids etc., one of the basic operations is building geometric structures. Geometric structures are used in Solid Modeling and In particular in Constructive Solid Geometry [Requicha 19801, from which the language PADL-2 [Brown 1982] has been derived. Geometric frame structures are used also In robot programming languages such as AL or VAL [Bonner and Shin 1982, Lozano Perez 1983] (or expressing the kinematics of robot manipulator arms. Moreover, theoretical work on data abstraction [Guttag 1977] has led to the concept of the abstract data type in programming languages which has found applications in computer graphics [Mallgren 1982] and in computer animation [Magnetat-Thalman and Thalman 1985]. Action Modeling When modeling aotions, on the other hand, one of the basic oonoepts Is concurrency. Concurrency is obviously present in concurrent programming languages [Andrews and Schneider 1983] like concurrent PASCAL [Brinch Hansen 1975] and ADA. Another area were concurrency Is important is the slmuletion of parallel processes. Simulation needs tools for defining quasi - parallelism and for characterizing program entitles which may interact during a simulation. The first high - level tool to be developed was the class concept in SIMULA-67 [Dahl and Nyguard 19681, extending ideaa already contained in the ALGOL language. A class Is an object which is composed of declarations (variables, procedures and even classes) and statements. Many objects of the same class may be Instanced and let Interact during a eimulation. This concept was expanded in object-oriented languages like SMALLTALK [Goldberg and Robson 1983] which is provided with very powerful graphic capabilities. A similar approach is related to the concept of actor, proposed by Hewitt [Hewitt et al. 1973] in a typical Lisp - oriented research environment. Hewitt defined an actor as an object that can send and/or receive messagea. Programming an actor system consists merely of telling the different classes of actors how to formulate their messages and how to respond to the meseages they receive. The actor concept produced several languages like PLANNER, PLASMA (Hewitt and Smith 1975], and ASAS [Reynolds 1982]. Another aspect of action modeling can be found in taak-level robot programming20. The languages of this type (for example, Autopass [Lieberman and Wesley 1977] and RAPT [Popple- stone et al. 1978]) are based on classic techniques of problem solving and planning [Nlllson 1982]. NEM is influenced by the actor concept and is based on data abstraction. However, even in the limited context of an "actor world1', it does not want to be general purpose: It is aimed at human movements. The rationale is that the actor concept, In Itself, Is too general and It is meaningful to make explicit the specific representation features that are typical of the human body and not of a car or a star-ship. Moreover, NEM Is an experimental tool, I.e. an 20 Robot programming is conventionally classilied in (a) actuator isvel, (b) robot or arm Itval, (c) task Itval. In (ha first case a program is a sequence ol commands to the joint actuators, in the second oast it is a sequence ol commands to the end-effector, and In lha test case it is a description of the goal.

NEM 155 environment in which it is possible to formulate and test hypotheses about motor control and motor knowledge.

This Page Intentionally Left Blank

PART II ENVIRONMENT This part contsins two chapters which deal with two apparently unrelated aspects of the actor/environment relation: Chapter 1: Compliance Chapter 2: Cognitive Modeling of Purposive Action The first chapter proceeds in a "bottom-up" way: It looks at the physical interface between an actor and his environment. When we carry loads, when we hit against obstacles, when we exert a force against some object or when we manipulate it, we must be careful about forces and constraints that we know only within large error margins. A machine that faces problems of this kind must be compliant, that is its motor apparatus must be able to exchange energy with the environment in both ways: absorbing and releasing, like a cane. However, this is not enough: Many machines are compliant in a specific way (think, for example, of the shock absorbers of a car) that can be appropriate for a limited class of problems only. What is needed is the knowledge of compliance that can make motor planning able to exploit it. Modeling is the first step towards knowledge and the chapter investigates a representation of compliance that may be "cognitively penetrable". The second chapter proceeds in a "top-down" way, looking at the problem of the actor/environment relation from the planning viewpoint. There are at least three main ingredients In action planning : (i) representations of skills, (ii) representations of the working environment, (iii) representations of the goal, and (iv) methods to achieve it. Classical methods of Artificial Intelligence mostly face the last issue only. The chapter reviews them and discusses the kind of knowledge that is needed in action planning, contrasting propositional vs. analogical knowledge. Finally, two specific cases are investigated in more detail: navigation and manipulation.

This Page Intentionally Left Blank

PART II Chapter 1 COMPLIANCE F. A. Mussa Ivaldi CONTENTS

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasco (Editors) © Elsevier Science Publishers B.V. (North-Holland), 1986 161 COMPLIANCE F.A. Mussa Ivaldi The implications of the muscle spring-like properties in the control of multi-joint arm posture and movement are considered. To this end some of the geometrical aspects are introduced which constitute a major novelty with respect to single joint studies. In particular, the serial and parallel connections of spring-like actuators are examined together with the non-linear factors in the control of multi-joint posture and movement. An experimental approach to the measurement and representation of the hand stiffness is presented and its implications are discussed in relation with the independent vs. synergistic control of muscle stiffness. 1. INTERACTION WITH THE ENVIRONMENT 1.1. Introduction When we make a movement, we interact with the environment in several ways: we may carry loads, we may exert a force against some object or we may hit obstacles. The act of manipulating has been defined, in a suggestive dramatization, as controlling the hand through a sequence of "collisions with unwanted forces" [Goertz 1963]. These forces may arise from contact with external constraints whose location in space is known within an error margin when movements are planned. Dealing with such "real life" movements is one of the most advanced topics in motor physiology and today's robotic research. In fact, current industrial manipulators are used for tasks involving predominantly free-motion, such as welding and spraying. The use of robots in assembly processes is limited by the complexity of the interactions between manipulator and workpieces. The problem here is that when the hand effector contacts a workpiece, the latter becomes an integral part of the system to be controlled [Hogan 1962]. As a consequence, the control parameters must be adapted to each particular task in order to generate the desired behavior. Similar problems are encountered even in free motion of multi-joint manipulators since large inertial forces, depending on arm configuration and on the products between joint velocities,

 limit the efficacy of linear feedback techniques to relatively slow movements [Raibert and Horn 1978, Benati et al. 1980). Here the inertia of the arm acts as an unwanted perturbation when linear feedback control schemes are adopted. A way to control the inertlal interactions occurring during free motion is to explicitly compute the actuator torques which drive the hand along the desired path, i.e. to solve, in real time, the so-called "inverse dynamic problem". This option, which has recently been made available by recursive computational techniques [Luh et al. 1960, Hollerbach 1980], is only partially satisfactory as the problem of insuring dynamic stability, i.e. of maintaining the hand along the desired path in spite of perturbations occurring during movement, remains unsolved. A main aspect of versatile, or dexterous, manipulation is the interaction between high level task planning processes and low level control processes. This is a two-way interaction. For example, the response of the manipulator to impacts with external constraints may be the business of the task planner whereas the choice of a particular trajectory in the workspace has implications for the control laws. The interactions between high and low level processes, however, don't preclude modularity in the organization of the motor control system. A clear distinction between planning and control (or between central and peripheral processes) is, indeed, a prerequisite for handling the complexity of motor behavior. The interrelation between these processes may be described as an exchange of information which takes place at their interface. For example, consider the task of tracing a line with a pencil on a rigid, curved surface. The surface is a constraint both in the real movement space and in the planning space, since movement can occur only along planes tangent to it. If the manipulator is represented as an ideal massless object, forces can only be generated by reaction with the constraint, i.e. orthogonally to the directions of movement freedom. The geometry of the external constraints may be used to decompose the task description in terms of motion (tangent to the constraint surface) and forces (orthogonal to the constraint surface). The idea of partitioning the task space into subspaces where force and movement tasks are reciprocally defined was proposed by Paul and Shimano [1976] and formalized by Mason [1979] for robot manipulators. From the description of the constraint surface Paul and Shimano derived a control scheme which, on each servo cycle, assigns each joint either a position or a force control law. In a similar approach, Raibert and Craig [1981] proposed a hybrid controller, where each actuator is, at the same time, force and position controlled in such a way as to guarantee exclusive force or position control of the end effector along each axis. The underlying Idea is that a clear separation between force and position control must be defined in end effector coordinates rather than in actuator coordinates. A significant consequence is that the controller must explicitly take care of the geometrical transformations between end effector and actuator variables in order to close both the force and the position feedback loops. Here, a major assumption is that, on the basis of the geometry of the external constraints, a choice must be made between force and position control. Alternative solutions have been formulated which are based on a continuous regulation of the impedance of the terminal device [Hanafusa and Asada 1977, Salisbury 1980, 1982, Hogan 1980, 1982, Benati et. al. 1980].

Compliance 163 As will be discussed later in this chapter, stiffness expresses the relationship between forces and displacements in static conditions. A definition of the stiffness is given by Hooke's law describing the behavior of a iinear spring: (1) F = K ■ {X - X0) where X is the actual position of the spring, X. is its equilibrium position, K is the stiffness constant and F the resulting force. Eq. (1) may be implemented by a feedback system with an input displacement and an output force. The compliance, C is defined by the inverse law: (2) X = C ■ (F - F0) where C = Kl and Fq — —K ■ Xq. Here, an input force causes an output displacement. Ideal force control corresponds to the selection of infinite compliance (zero stiffness), while ideal position control is obtained at infinite stiffness (zero compliance). Between these two extremes lies the idea of stiffness control (or, more generally, impedance control). In a stiffness control scheme, the task planner defines the stiffness of the end effector in different directions, in accordance with the geometry of the constraints and the task requirements. Then, the actuator stiffness is derived which implements the desired hand effector stiffness by means of geometrical transformations. The novelty, with respect to pure position control, is not in the implementation of (1) at the actuator level. Actual position controllers are in fact implemented by means of proportional feedback loops with finite gain. The difference, in stiffness control, is that the planner itself dictates the value of K to the controller according to the demands of the task. Again, the central issue is the exchange of information between high and low level processes. 1.2. A Physiological Background We have introduced the concept of stiffness control as an alternative to pure force or position control. Let us discuss the relevance of these ideas in the biological domain. Stiffness control requires elastic actuators with variable stiffness parameters. They may be implemented either by local feedback, which makes them "look like" springs, or by structural elasticity. The latter solution, which has been adopted in remote center compliance (RCC) devices (Watson 1976, Drake 1977, Whitney 1982], is only applied in robotics for the execution of specific tasks, such as inserting pegs into holes. This limitation results from the fact that currently available elastic

 materials have fixed rather than "tunable" spring constants. In contrast, muscles have elastic properties which depend on the level of alpha-activity [Rack and Westbury 1969]. The isometric force/length relationship in individual muscle fibres was studied by Gordon et al. [19661 who related the development of tension at different sarcomere lengths to the degree of overlapping between actin and myosin filaments.1 According to the sliding filaments theory of muscle contraction [Huxley 1963, 1969], the force generation process is caused, within the muscles, by the physical interaction between myosin and actin filaments. This interaction Is provided by the heavy - meromyosin (HMM) molecular subfragments originating from the myosin filaments. The subfragments HMM-S1, under the influence of the action potentials propagating along a fibre, form covalent bonds, or cross - bridges, with the actin filaments. Such bonds are thought to provide both the contractile force and most of the stiffness in the activated muscle. Hence, the increase of muscle stiffness, observed with increasing alpha - activity, is considered a direct consequence of the generation of new cross - bridges. However, the functional role of muscle elasticity has been until recently underestimated by most motor physiologists. Muscles were only considered as "force generators" and limb stability was seen exclusively as a consequence of the reflex activity caused by external disturbances. This point of view was first challenged by Russian investigators [Asatryan and Feldman 1965, Feldman 1966] who formulated the hypothesis that the Central Nervous System (CNS) controls muscle spring-like properties, i.e that the neural activity reaching the motoneurons from higher structures Is specifying neither length nor tension but their functional relationship. This hypothesis accounts for the initial force compensation occurring during fast locomotion [Qrlllner 1972]. The relative contributions of segmental reflexes and muscle stiffness to load compensation during centrally initiated head movements was experimentally evaluated by Bizzi et al. [1978] in chronically vestibulectomized monkeys. By comparing loaded head movements before and after transection of the cervical dorsal roots, It was concluded that reflexes contributed for not more than 10-30 % of the observed compensation. Another set of experiments [Polit and Bizzi 1978] with deafferented monkeys demonstrated that goal directed elbow movements can be successfully executed without feedback intervention. These findings led to the conclusion that movements are centrally represented as changes in the equilibrium position of the limbs defined by the interaction of antagonist muscles. Further evidence obtained with single-joint elbow movements strongly suggests that the equilibrium gradually shifts from initial to final position [Bizzi et al. 1982). The results of these experiments should not be taken to imply that segmental reflexes have only a minor role in the control of posture and movement. They demonstrate, rather, that control laws are implicitly encoded in the central commands, through the selection of length/tension functions. The Idea that spring-like behavior is relevant to motor control also led to the hypothesis that position and force feedback provided, respectively, by muscle spindles and tendon organs, act together to regulate muscle stiffness [Nichols and Houk 1976, Crago et al. 1976]. As a consequence, the overall behavior of the peripheral apparatus (muscles and ' In the structural organization ol vertebrate striated muscias, sarcomeres are the units which ara repeated longitudinally along the fibrils. These units consist primarily ol comb-like arrays ol overlapping actin and mycein arrays.

Compliance 165 reflexes) would be characterized by almost linear length tension curves in spite of the irregularities observed in isolated muscles and attributed to the sliding filaments mechanism [Joyce et al. 1969]. Once the idea that the elastic properties of muscles are relevant to motor control is accepted, we are still a long way from understanding how real life movements are generated and controlled by the CNS. Starting from the background provided by single-joint studies, the remaining paragraphs are dedicated to illustrating how some geometrical concepts may be applied to approach the problems of multi-joint arm control and to represent the total elastic behavior of the arm under the influence of spring-like actuators. The ultimate goal of this chapter is to introduce the reader to some of the theoretical aspects and of the experimental techniques which have been developed recently to investigate the neural control of multi-joint arm mechanics. 1.3. Overview of the Chapter We will discuss the implications of the muscle spring-like properties in the control of multi-joint arm posture and movement. To this end we must introduce some of the geometrical aspects which constitute a major novelty with respect to single joint studies. In general, a spring-like system is defined as a system which stores potential energy or, equivaiently, as a system in which the force is related to the position through an integrable function. A linear spring, for example, when displaced from its equilibrium position, generates a proportional restoring force. This is known as Hooke's law of elasticity. In uni- dimensional systems, such as rubber bands or individual muscles, all the mechanical variables are scalar, whereas in multi - dimensional elastic systems, Hooke's law may be preserved in its simple form (force proportional to displacement) only if forces and displacements are treated as vectors. Consequently, stiffness, which is defined in a uni - dimensional spring as the ratio between force and displacement, in the multi-dimensional system must be considered as a tensor. The stiffness tensor relates an input displacement vector to an output force vector and is represented by a matrix. A key issue in vector and, more generally, in tensor geometry is what happens to these objects when the coordinate system in which they are observed changes. Force and position vectors can indeed be expressed in different coordinate systems. For example, one may express the displacement of the arm in terms of angular displacements at the joints, in terms of Cartesian displacement of the hand or in terms of changes in the individual muscle lengths. The point here is that whatever choice is made, one is observing the same physical system and the fundamental laws of mechanics must preserve their form in all descriptions. For example, the mechanical work is always given by the scalar product of force and displacement.

 In general, a complex system, such as the human arm, is made of smaller subsystems connected together. Motor coordination, in this sense, may be defined as the ability to control the whole system by acting upon its individual components. Hence, motor coordination is a geometrical issue, since it results In the mapping of mechanical variables between task-related coordinates and coordinates related to the individual actuators. Such mappings are derived from the geometry of the interconnections between the subsystems. They may be highly non - linear, as in the case of the transformation from joint angles to Cartesian coordinates of the hand, but they can be locally linearized when, instead of position vectors, small displacement vectors, or velocity vectors, are considered. When a coordinate transformation is linear, the position vectors are mapped into the new coordinates by multiplication with a constant matrix. This happens, for example, in the case of rotations. In contrast, when a mapping is non - linear, only the infinitesimal displacement vectors transform by linear multiplication with a matrix - the Jacobian matrix - whose elements are functions of the point where the displacements are taken. For example, a small rotation of the elbow generates different Cartesian displacements of the hand when it is applied from different initial elbow angles. Furthermore, the laws of mechanics establish some relationships between different variables, relationships which are used to derive other transformation laws. For example, the work produced during a displacement is numerically the same in all coordinate systems. Since the work is given by the scalar product of force and displacement vectors it follows that, in a coordinate change, the transformation of the force must compensate exactly the transformation of the displacement in such a way to keep the product invariant. This is stated in tensor calculus, more exotically, by saying that forces and displacements are tensors of opposite variance. They transform in the same coordinate change according to inverse laws or equivalent^, they transform according to the same law in opposite directions. Hence, the same Jacobian matrix is used to map a joint displacement into a Cartesian displacement of the hand and to map a Cartesian force at the hand Into a torque at the joints. Following similar arguments, it is possible to derive the transformation laws for the mechanical impedance. The latter Is defined as a tensor which generates an output force In response to an input movement variable. As previously shown, stiffness relates forces to displacements. Similarly, viscosity relates forces to velocities and Inertia relates forcea to accelerations. Starting from the invarlance of these definitions in all coordinate systems it is easily demonstrated that the mechanical Impedance during a coordinate change transforms in accordance with the same law followed by the force and according to the inverse law followed by the displacement. For example, the same Jacobian operator is used to transform Cartesian stiffness and Cartesian force at the hand respectively into joint stiffness, and joint torques. The direction of a mapping is a delicate issue, particularly when it occurs between spaces with different dimensions, as is often the case in biological systems. For example, the position of the hand in space Is completely described by six variables (three Cartesian coordinates and three orientation angles). In contrast, the independently controlled joint angles are at least

 seven. Furthermore, each joint variable is associated with a large set of muscle variables. At first sight, it may seem that we are facing the same kind of redundancy at the muscular and the skeletal levels: many joints and many muscles control few kinematic variables associated with the hand. We will show, however, that the interconnections of the muscles and of the joints are radically different and that they generate complementary mapping rules. The muscles acting around a joint are connected to the skeleton in a "parallel" configuration. In contrast, the skeletal segments are connected together in a "serial" chain. Let us discuss these two situations by considering two simple mechanical examples which allow us to simplify the mathematical details without affecting the main issues. The parallel connectivity is represented by a set of linear actuators attached to a common end effector. In contrast, the serial configuration is depicted by a set of actuators attached to each other along an open chain. The free end of the chain is considered, in this case, as the end effector. The excess of dimensions (i.e. of independently controlled variables) of the actuator space with respect to the end effector space generates, in both the serial and the parallel configurations, three possible kinds of mapping, namely: 1. Well defined mapping, when a vector in one space transforms uniquely into a vector in the other space. 2. Under - constrained mapping, when a vector in one space maps into an infinite set of vectors in the other space. 3. Over-constrained mapping, when a vector in one space may not be transformed into any vector in the other space. These three cases are schematically summarized In Table I. The problem of dealing with ill- defined transformations in the biological domain has been recently brought to attention with respect to the processes of early vision, where the information about the shape of objects is recovered from limited and noisy sets of input data [Poggio and Torre 1984]. The underconstrained mapping results from the lack of equations with respect to the number of vector components in the target space. In contrast, the overconstrained mapping reflects an excess of these equations. The example of a well defined mapping is, in the parallel connectivity, the transformation of the displacement from end effector to actuator 'coordinates, whereas, in the serial arrangement, the transformation for the displacements is well defined in the other direction, from the actuators to the end effector. From our discussion, on the reciprocity of force and displacement transformations, it follows that the force mapping is well defined going from actuator to end effector coordinates in the parallel connections while in the serial connections it is well defined the other way around. Hence, independently of the redundancy of the musculoskeletal system, it is always possible to derive a unique solution for the following transformations:

 a)- DISPLACEMENTS Connectivity Parallel Serial Actuators >-» End Effector Over-constrained Well-defined End Effector >—> Actuators Well-defined Under-constrained b)- FORCES Connectivity Parallel

Serial Actuators >-» End Effector Well-defined Over-constrained End Effector >-> Actuators Under-constrained Well-defined Table 1: Transformation schemes lor parallel and serial connectivities, a)- mapping ol displacement vectors; b) mapping ol force vectors. - from joint angular displacements to muscle length changes - from joint angular displacements to hand Cartesian displacements - from muscle tensions to joint torques - from hand forces to joint torques When the actuators have spring-like properties, further transformation rules are formulated for stiffness and its inverse (compliance). We have seen in fact that stiffness follows the same mapping rules as force. Thus, mapping from muscle to joint stiffness is well defined. By similar arguments it can be derived that compliance transforms as do displacement vectors2. When one tries to invert the direction of a well defined mapping between spaces of different dimensions, one obtains either an overconstrained or an underconstrained transformation. In parallel configurations the mapping of displacements from actuators to end effector coordinates is overconstrained while the mapping of forces in the other direction is underconstrained. In the serial arrangements the situation is exactly reversed: the end effector displacements transform into an infinite set of actuator displacements whereas the transformation from actuator to end effector forces Is overconstrained. In the case of the redundant musculoskeletal system these statements are summarized as follows: 1. An arbitrary set of muscle - length changes may be incompatible with the actual connections of the muscles to the skeleton. Therefore, it doesn't necessarily map into any possible joint angular displacement. * When dealing with matrices, such as compliance and stillness, some extra precautions must be taken. A matrix may in fact become singular, i.e. non-inverlible, in the transformation to spaces with more dimensions, even it the mapping is unique. This is the case, lor example, when the joint compliance is mapped into the muscle compliance matrix.

 2. When there are more degrees of freedom at the joints than at the hand, an arbitrary displacement of the hand in space maps into an infinite set of corresponding angular displacements at the joints. 3. A given net joint torque vector may be achieved by an infinite set of muscle tensions. 4. An arbitrary set of joint torques doesn't necessarily map into any Cartesian force at the hand. (For example, when the forearm is fully extended, it is not possible to find a force at the hand which generates torques of opposite sign at the shoulder and at the elbow). When a mapping is underconstrained, a vector

transforms into an infinite set of vectors. For example, a desired joint torque may be generated by an infinite number of force patterns in agonist/antagonist muscle pairs. These patterns lie on a common plane (more precisely, hyper-plane) in the space spanned by the actuator forces. The problem of locating a particular solution, i.e. a particular point of the solution plane, is often approached by imposing further constraints on the transformation. This is equivalent to adding extra equations to an underconstrained algebraic system in order to make it well behaved. A well known constraint is that of imposing that the desired point in the solution plane be also at the minimum distance from the origin. This yields the "minimum norm solution" which is obtained by a linear operator known as the Moore-Penrose generalized inverse. In the example of force distribution among muscles acting around a joint, it corresponds to the choice of minimizing the sum of the squares of the muscles' forces. So, if the desired net torque is zero, all the muscles must exert zero force in order to satisfy this extra constraint. This simple example should be sufficient to rule out the hypothesis that the CNS actually adopts such constraints to implement motor coordination. It is well known, in fact, that zero joint torque is commonly generated at different co - contraction levels of agonist/antagonist muscle pairs. Furthermore, the fact that muscles have spring-like properties represents, in itself, a natural constraint to the distribution of forces. For any given set of actuator stiffnesses, the distribution of force, in a parallel connection, must satisfy the physical law of minimizing the stored potential energy. Similarly, in a serial configuration, the end effector displacement maps into a unique displacement of the actuators when the actuator compliances are given. If the tip of a serial kinematic chain is displaced, all the elements will displace in such a way to keep the potential energy at a minimum. The spring-like properties of the musculo - skeletal apparatus then act as a constraint which yields a well defined mapping of force and position variables across spaces of different dimensions. This conclusion holds in general for all the components of the motor impedance, such as inertia and viscosity. Impedance, in fact, provides a linkage between force and movement variables. Since these variables transform according to inverse laws, a well defined transformation between task and actuator variables is always possible if the impedance is known. This is consistent with the simple observation that when we displace the tip of a multi - articulated physical object, all its parts move in a well defined manner without bothering about kinematic redundancy! Thus, the concept of motor redundancy is a purely kinematic definition which must be re

- examined when dynamics (i.e. the relationship between forces and movements) is taken into account. When the arm interacts with the environment, not only the kinematic parameters but also the impedance components are relevant to motor behavior.

 Whether and how the CNS controls the motor impedance of the arm by "tuning" the spring-like properties of the muscles in multi-joint posture and movement is entirely a matter of experimental investigation. A major problem to solve in any experimental approach to this issue is that of measuring and representing the net effect of the muscle spring - like properties on the global behavior of the arm. In other words, the stiffness must be observed in a task-related coordinate system. This goal was achieved in a recent investigation [Mussa Ivaldi et al. 1985] which is discussed in detail in sect. 4. The hand stiffness was measured, in planar posture, according to its own definition. Subjects were requested to maintain the hand in a position of the horizontal plane while holding the tip of a two-link manipulandum. The manipulandum was used both to monitor the hand position and to apply servoed - displacements of small amplitude along several directions in the plane. When a displacement was applied, the subjects generated a restoring force pulling toward the original position. This force resulted from the elastic properties of the muscles and, to some extent, from reflex activity but not, at least within a certain time margin, from any voluntary contribution. The experimental procedure consisted of measuring several displacements and restoring force vectors and of estimating the hand stiffness matrix by linear least-squares regression on the equations which express Hooke's law of elasticity. The advantage of this method, with respect to those based upon the analysis of the dynamic responses to force perturbations, is that of being totally independent of any assumption about arm dynamics. Once the hand stiffness matrix was measured in a workspace location, it was used to derive the joint stiffness matrix, relating joint torques to angular displacements. The knowledge of joint stiffness represents a further step towards understanding the organization of muscle activity in multi-joint posture. For example, in these experiments it was found that the hand stiffness matrix was symmetric in all tested workspace locations. This means that a displacement along the y-axis generated a force component along the x-axis equal to the force component along y generated by a displacement of the same amplitude along x. It has been proved that this kind of symmetry is a necessary and sufficient condition to state that the force field is conservative and that a scalar potential function, which represents a storage of elastic energy during a displacement, may be defined in the workspace. It has also been demonstrated that the symmetry of the stiffness is preserved when this matrix Is transformed into joint coordinates. That is, a displacement of the shoulder generates an elbow torque equal to the shoulder torque generated by a displacement of the same amplitude at the elbow. Since the reflexes were included in the measurement of the stiffness, its symmetry implies either the irrelevance or the perfect balance of the cross coupling reflexes between muscles acting around different joints. These reflexes cause, say, a change in tension in the elbow muscles in response to a stretch of the shoulder muscles and vice versa. If the gains along the two pathways (shoulder to elbow and elbow to shoulder) were different, the joint and the hand stiffness matrices wouldn't be symmetric and multi-joint posture couldn't be characterized by a spring-like behavior. In contrast, our findings rigorously demonstrated the spring-like nature of hand posture, a fact that couldn't have been derived from single joint studies.

Compliance 171 Apart from the previous consideration about symmetry, the expression of the stiffness as a table of numbers is not the optimal way to capture its regularities as the hand position changes in the workspace. An alternative representation was obtained by plotting the locus of the restoring force vectors at each equilibrium position when a rotating displacement of constant amplitude was imposed as an input to the stiffness matrix. The outcome of this process is a force - ellipse, whose parameters directly relate to the stiffness matrix. The ellipse, indeed, is oriented with the major axis in the direction of maximum stiffness and its size is proportional to the determinant of the stiffness matrix: it represents the global amount of stiffness along all directions. Another important parameter is the aspect ratio, or "shape" of the ellipse defined as the ratio between the major and the minor axes. A ratio equal to one - that of a circle - corresponds to an isotropic stiffness. At least in theory, the CNS has the potential of modulating the stiffness shape, size and orientation by independently "tuning" the individual muscle stiffnesses. However, our results did not support this hypothesis. When subjects were asked to maintain hand posture in different workspace locations, a repeatable and consistent pattern of stiffness shape and orientation was found: at each location, the stiffness ellipse was oriented approximately along the line joining the hand to the shoulder and the shape was more isotropic, or circular, in the proximal than in the distal positions. The in variance in shape and orientation of the stiffness measured in each position was preserved even when the subjects had to maintain hand posture against an externally applied force of constant orientation. However, in this condition the stiffness size increased by more than 100%, reflecting an increased neuromuscular activity. The observed invariance of the stiffness shape and orientation reflects a constraint which must be satisfied by the distribution of neural activity to the muscles. In fact, only a uniformly distributed change in muscle stiffness can cause a variation of hand stiffness size while leaving the other parameters unchanged. Another experimentally observed constraint was the geometrical pattern followed by the stiffness in the workspace. A subsequent work [Flash and Mussa Ivaldi 1964] derived a condition which must be satisfied at the joint level was derived from the orientation of the stiffness ellipses along the line joining the hand to the shoulder. It was demonstrated, and experimentally confirmed, that to produce this pattern the shoulder joint stiffness covaries in the workspace with the stiffness term provided by the double joint muscles. Taken together, the outcomes of these experiments suggest that muscle stiffness is synergi- cally controlled. As a consequence, the pattern of hand stiffness in the workspace remains invariant in shape and orientation in spite of changes in the global muscle activity. In accordance with this hypothesis, the Cartesian stiffness of the hand can still be modulated by changing the configuration of joint angles adopted to achieve a desired hand position in the workspace. Since the stiffness, together with the other components of the impedance, expresses the reaction to external perturbations, the choice of a particular arm configuration may be dictated by the interactions with the environment involved in a motor task. The following sections are dedicated to a more detailed discussion of these issues.

 2. GEOMETRIC ASPECTS OF MOTOR COORDINATION 2.1. General Considerations An ordinary linear spring is a system described by Hooke's law (1): F = K-(X-X0) which states that, when the spring is displaced from its equilibrium position, a force is generated proportional to the amount of displacement and in the opposite direction (K is then negative). In uni - dimensional systems, like rubber bands, all the quantities involved are scalar. When dealing with multi - dimensional elastic systems, (1) may be preserved in its form but forces and positions must be treated as vectors: (3) F = K ■ (X - X0) with (F = [F..F,, ...,Fn]T W j X = [Xu

Xt, ...,Xn}T [X0 = [Xq,, X0j, ..., X0n]T This equation expresses a transformation between n-dimensional vectors and, as a consequence, K is a square nxn matrix - the stiffness matrix. Linear elasticity is not required in order to define the stiffness matrix. Stiffness can, in fact, be defined, at any point, as a differential quantity which relates a displacement dX to the consequent variation of force dF. Accordingly, the elements of the stiffness matrix K are given by: In the proximity of the equilibrium position of the system, XQ, (defined by Fpy = 0), (3) becomes (6) F = K ■ dX Force and position vectors may be expressed in different coordinate systems. For instance, in a coordinate change the position vector X is transformed into a vector X' whose m components are functionally related to the n components of X, i.e: (7) •*1 ~ Xl(LXi,X2,...,Xn) X'j = X'^Xl, Jf2i • • • > Xn) X'm = X'm(Xi, Xit..., Xn)

Compliance 173 The fact that m and n may be different means that the same system can be observed in spaces of different dimensions. For example, the behavior of the arm can be described in terms of hand coordinates or in terms of muscle-length coordinates. Even if the coordinate transformation (7) is non - linear, the corresponding transformation for the infinitesimal displacement vectors is linear. Differentiating each function in (7) we obtain: (8) where, dX\ = J\,\ dX\ -r J|,2 dXi + • • • + Ji,n dXn dX[-— Ji,\ dXi + J2,2 dXi H h Ji,n dXn dX'm = Jm,i dXx +Jm,2dX2-\ \-Jm>ndXn (9) J..= «S A> dXj In vector-matrix notation, the system (8) becomes: (10) dX' = J • dX The matrix J is called the Jacobian of the transformation. The values of the partial derivatives which constitute its elements, may or may not depend upon the position X where they are taken. If (and only if) ihey don't depend on position, the transformation (7) is linear. If m = n, J is a square matrix and an inverse transformation (11) dX = J1dX' can be defined, provided that dct(J)^0 The transformation of the force vector in a coordinate change can be derived by making use of the physical principle according to which the mechanical work generated during a displacement is invariant with respect to coordinate transformations.3 The work, defined as the inner product of force and displacement, is given, in the original coordinate system, by: (1a) dW = FT ■ dX whereas, after the transformation, we have: (13) dW = F,T • dX' 3 This is a restatement of the well-known principle of virtual works which is the starting point for the Lagrangian formulation of mechanics (Goldstein 1965].

174 F.A Mussa Ivaldi By equating (18) and (13) and substituting for dX' its expression in terms of dX (10) one obtains: (14) F = JT • F' which gives the force in the original coordinate system as a function of P. By comparing (10) with (14), we observe that force and displacement vectors transform in opposite directions by means of the same Jacoblan operator. This is also a well known result of tensor calculus applied to mechanics [Brillouin 1984): Forces and displacements are tensors of opposite variance since their product is a true scalar, or invariant. Consequently, they transform according to inverse laws. Finally, I would like to consider, in the context of coordinate transformations, stiffness, compliance and, more generally, the motor impedance. Again, the the transformation laws are derived from the invariance of physical laws. In the case of stiffness, its own definition (6) must remain the same in all coordinate systems. Thus, (15) F = JT ■ F' = JT • K' • dX' = JT • K' • J dX = K • dX i.e., (16) K = JTK'J As for the force, the stiffness in the new coordinates is the Input and the stiffness in the original coordinates is the output of transformation (16) which may not be invertlble. When compliance, C, is considered, the corresponding transformation (17) C' = JCJT is derived by assuming that the definition (1fl) displacement = compliance x force remains valid in all coordinate systems. With the same method we can also derive the transformation laws for the other components of the mechanical impedance of a multidimensional system. For example, the inertia I' defined by: (19) F' = I' • X« becomes, after the coordinate change, (80) i = jT-r-j

Compliance 175 This equation, however, is only correct at zero velocity since (21) *=£(*) = £(J-*)=i-*+J-* and the first term to the right is equal to zero only if the transformation is linear (i.e. if J - constant) or if X - 0. An intuitive interpretation of the transformations presented in this section will now be illustrated with two elementary examples which allow us to deal with the basic geometrical problems without tedious algebraic operations. The main conclusions to be drawn from these examples apply also to a system as complex as the human arm and provide some insight into the problem of controlling and utilizing "motor redundancy". 2.2. First Example: the Parallel Connection The simple system of figure 1 is composed of two spring-like actuators, A. and A,,, attached to the same point E - the end effector - in a parallel configuration. There are two distinct approaches to such a system: either we have access to the individual variables (forces and lengths) associated with each actuator, or we observe only the behavior at E. In the first fSnmm A Al -sS&mmN.r- A2 ft re h m/mrr^smsmsuLr^ si -p*<y *a Figure 1. parallel connection of two spring-like actuators (A1 and A2). A) Configuration (E - end effector), B) Actuator variables (fl, 12: forces, x1, x2: positions) , C) End-effector variables (F: force, X: position).

176 F.A. Mussa Ivaldi case the system is described by a position (or "length") vector <22> x = |*i.*»F and by a force (or "tension") vector (23) f=|/i,A]T as indicated in figure 1-b. The second approach (figure 1-c) corresponds to the choice of observing the position X of the end effector and the force F resulting from the combined action of both actuators4. Our interest here is to explore the relationships between the two approaches. The parallel connection between the actuators is expressed by a "common position" constraint, i.e.: (24) \xx = X K ' S x2 = X-D If, instead of the position variables x-, x« and X, the corresponding displacements are considered, (24) becomes: (25) I ixt - &X 6x2 — 6X I.e., from the measure of the displacement at the end effector, E, the displacement of the actuators A1 and A2 is derived5. In contrast, any two independent values for the displacements of the actuators, may not be compatible with the system constraint (24). The situation is visualized in figure 2-a, where the two axes represent the displacements of A. and A,. If the actuators were disconnected, it would be possible to displace them independently. In other words, every point in the plane of figure 2 - a would be accessible, at least within a boundary. In contrast, the actuators are actually connected to the same point. This fact may be expressed as a constraint on the displacement variables (A. and A, can only be displaced by the same amount). Consequently, the only points which may be reached, in the actuator displacement plane, lie on a line which represents the common position constraint. In vector notation, (25) becomes: (26) Sx = J ■ 6X * X and F should be considered not as scalar quantities but as uni - dimensional vectors. The difference Is subtle but Important: a true scalar is, by definition, invariant for coordinate changes. Examples ol such an entity are work and energy. In contrast, lorces and positions, even when expressed as single real numbers, do changa in coordinate transformations. 5 The symbol delta is used here to indicate Unite displacements. When the geometry ol the system is non - linear, it should be considered as indicating small displacements, in order to remain within linear approximations.

Compliance 177 n re Xp ixi Figure 2. Parallel connection, a) Actuator displacement space (tangent space). The dashed line represents the actuator displacements possible in the geometry ol the connection b) Actuator lorce space. The dashed line represents, lor a value F ol the end-eflector force, the solutions lor the actuator force. P minimum norm solution where J is the Jacobian of the system (27) J = [Uf Since J is not square, it cannot be inverted and since it has more rows than columns the system (26) is overconstrained, i.e. there may be no solutions given an arbitrary vector of dispace- ments. When the end effector displacement has to be derived from the actuators displacements, (26) is a particular case of an algebraic system with more equations than unknowns. The same kind of situation is encountered when one has to estimate few unknown parameters of a given function from a large set of measurements of the dependent and Independent variables. A typical solution, in this case, is the least squares approximation, i.e. given the fact that the experimental data do not satisfy exactly the hypothesized functional relationship (because of the noise), the parameters are chosen in such a way to render the function "as close as possible" to the experimental data. The pseudo - inverse, or Moore - Penrose6 generalized inverse P, defined as (28) P = (JTJ)"'JT = 2 2 More details on the generalized inverses may be lound in Ben - Israel and Grevilla, 1974.

178 F.A. Mussa Ivaldi gives the "least squares" solution for SX (29) tfS[}}]x 6xi - (Sxi+Sxj) If one substitutes this value into (26), a vector is obtained which is at the minimum Euclidean distance from the original displacement vector, while satisfying the constraint of the parallel connection. Let's now consider the force vectors. By applying the general rule (15) derived in the previous section we obtain: (30) F = JT ■ f = /, + /, which is, in this case, an obvious result. As pointed out, the direction of the mapping, obtained by J, is different for forces and displacements: Given a displacement of the end effector, it is possible to derive the displacement of the actuators (eq. 26) and given the force at the actuators it is possible to derive the force at the end effector (eq. 30). The inverse transformations are not defined since J cannot be inverted. For any given end effector force (30) defines a straight line in the plane of the actuator forces (figure 2-b). More generally, for n actuators it defines a hyperplane in the n - dimensional space of the actuators forces, for each value of the end effector force. Here is another aspect of the dual relationship between force and position vectors: while (25) Is an over-constrained system for X - i.e. there may be no solution - (30) is an under -

constrained system for f - i.e. with infinite solutions. From the solutions for f it is possible to choose the one which has the minimum norm, i.e. which also satisfies the constraint: (31) ‖f‖2 = fl + fl = minimum This is done by utilizing the pseudo - inverse matrix P: r w Pi7" (31) f = PT . F = - - 1 minimum norm — * * 0 0 The same matrix P is used, for positions, to derive the least square solution to an over- constrained system and, for forces, to derive the minimum norm solutions. This completes the picture as regards duality between forces and displacements. So far, we have not considered the fact that the actuators have spring-like properties, i.e that forces and displacements are related to each other through integrable functions. In the simple case of linear elasticity we have : (33) i Jl = *>•(*!-*<>,) tf: ki ■ (i, - x0,)

 where k1 and kg are the elastic constants and Xq and Xq are the rest lengths of A. and A,,. Using the vector/matrix notation, we write: (34) with (35) f = k • (x - x0) k = it, 0 0 k2 *o = [x0l,xQl\T The stiffness k may be defined, more generally, as a local differential operator by: (36) <5f = k*x The transformation for the total stiffness, K, at the end effector is given by (16)' (37) tf = JTkJ = [l,l] ki 0 0 ki *i + k2 Once more, this equation cannot be inverted since J is rectangular. One could be tempted to overcome this problem by considering the compliance instead of the stiffness. Given the compliance C of the end - effector, we have (eq. 74): c = J-C-JT and, since (38) k = c-» the problem seems to be solved. Unfortunately, a basic theorem of matrix algebra states that the rank of a matrix product cannot exceed the smallest rank of its factors. Thus, the rank of c, as it is derived from (17), is not greater than one. It follows that c cannot be inverted, as it is required in (38), to derive k. The sad moral of this story is that there is no way to overcome the redundancy of the system without adding further constraints. An example of such a constraint is the minimum norm solution (32). This would also provide a unique solution for the actuator stiffness given the total end effector stiffness. Is this an acceptable solution when the actuators have spring-like properties? To answer this question we must consider that (32) implies that f-0 if F«0. On the contrary, when FaO, it is (39) f,F=0 = fe = kt+k2 (x0l - x0l + D) 1 -1

 So, the minimum norm solution is not acceptable for this system unless: (40) xo, =xo,-D i.e. unless the rest length of each actuator coincides exactly with the equilibrium position of the end effector. However, the minimum norm hypothesis can be slightly modified by a translation in the actuator force plane: (41) f = f - fe It follows that (4a) F = jTf = jTf' + jTf. = JTf since the "bias" component fe is not generating any end-effector force, i.e. (43) JT fe = 0 Eq. 27 could still admit a minimum norm solution for f. Such a solution gives the condition: (44) ki = k2 i.e. a uniform distribution of stiffness between the actuators. Figure 3 summarizes the solutions for the actuator forces given the end effector net force and the actuators stiffness matrix. The dashed lines represent the solution planes for f given (45) JT • f - F without any other constraints. The vector I, given by the elastic parameters of the actuators, is a constant bias representing the "internal force" of the system. The minimum norm solution moves, as the total force changes, along the solid line (k.«k,,), orthogonal to the dashed lines. In general, given the stiffness matrix k the system (42) admits a unique solution. This is obtained starting from the stiffness definition (46) f = k • <5x' where Sx' = x - x# The position vector - x , is such that (47) JT-f(xc)=0

 kl < k3 \ \ \ ce-« < n < Fa Figure 3. Parallel Connection. Force trajectories for spring-like actuators at different stilfness values. Dashed lines: solution planes for three values of the end effector lorce . and defines the equilibrium point of the end effector. Substituting (36) in (46) and introducing the definition of the end effector compliance (48) 6X = CF we obtain: (49) f' = Qk • F with (50) Qk = k • J ■ C This is a particular case of the weighted pseudoinverse: (51) QA = AJT.(JT-AJ) ' with A « k. The choice of Q» as a pseudoinverse of J corresponds to the additional constraint of minimizing the quadratic form: (53) fT • A ' • f

 When A is proportional to the identity matrix, QA is the minimum norm pseudoinverse P that we discussed before. In the case of elastic actuators, (49) and (SO) are equivalent to the constraint of minimum potential energy, which is a physical law. In our simple case it is (53) Qlc = (fci + fc2)~ Then, for k1 > k,, the actual solution for the actuator forces lies below the minimum norm line while for k. < k,, it lies above it as indicated in figure 3. To summarize the description for the parallel connection, we make use in figure 4 of a linear graph notation. The arrows may be considered as "one way streets" leading from one variable to another through a linear transformation. The dashed line represents the pathway that must be followed in order to go from the force in the end effector space to the force in the actuator space. Once the target has been reached, the corresponding equation, in this case (49) and

(SO), can be obtained by multiplying the operators in cascade, as encountered along the path. The fact that the actuators have elastic properties has the important effect that each node in the graph is accessible from any other node. 2.3. Second Example: the Serial Connection Figure 5 is a sketch of another very simple system. It consists of two spring - like actuators A1 and Ag connected together in series. The end effector E is at the extreme right. Its position X is given by the sum of the actuators positions (54) X = xi + xj The corresponding equation for the displacements is: (55) SX = J • Sx with (56) J = [1.1] Using the same arguments as before, we obtain for the forces: (57) f = JT ■ F h = F ft = F

Compliance 183 '/" .N. XX ACTUATORS SPACE END-EFFECTOR SPACE - ♦ v v • A u Figurs 4. Top linear graph rspresentation of the parallel connection (explanation in the text). Bottom: linear graph notation. / At A2 ^^mms^-r-^xxmimsj^ xl ->• x2 Figurs 5. Serial connection of two spring-like actuators.

184 F.A. Mussa Ivaldi which is a "common force" constraint. By comparing these equations with (25), (26), and (30) of the previous paragraph, it is evident that in the parallel connection the roles played by forces and displacements are exchanged with respect to the serial connection. (If we substitute forces for displacements and vice versa in (54), (55) and (57) we obtain the equations which describe the parallel arrangement.) This is enough to conclude that the serial system is completely described by exchanging forces for displacements in all the considerations of the previous section. Accordingly, also the stiffnesses have to be replaced by compliances and vice versa. Let us focus briefly on the problem of inverting (55) in order to obtain the displacement vector x given the end effector displacement. Again, as in (30) of the previous section, we have an underconstrained system consisting of one equation and two unknowns. If, however, the actuators compliance c is given, i.e. if (58) 6x: with c, 0 0 c2 it is possible to write the displacement vector as a function of the end-effector displacement by substituting (57) for f and considering that (59) F = K-6X where Finally, (60) with K = (Ci+c2)- Sx = Q'c • SX Qi=c-Jr-tf = (c,+c2)- This is the same kind of weighted pseudoinverse that was discussed previously. Both Qk and Q_ express the minimum potential energy constraint, the former in the force space and the latter in the displacement space. Whitney [1972] proposed the use of the pseudoinverse (61) Q'A = A-JT.(j-A-JT)"

Compliance 185 to solve the inverse kinematic problem for redundant manipulators, represented as serial kinematic chains. However, since he was dealing with non-elastic actuators, the choice of A was left arbitrary. In contrast, here, physical laws dictate a specific choice of A. Graphically, the situation is summarized in figure 6. The dashed lines represent the solution planes of (55), with no other constraints, for increasing end effector displacements . The fact that the actuators have spring - like properties is a constraint expressed by the statement that a solution should also lie along one of the three solid linas. When the compliance is uniformly distributed between the springs (c.=c2) we obtain the minimum norm solution, i.e. a line orthogonal to the solution hyperplanes. When the compliances of the actuators are not equal, the solution line is bent toward the axis of maximum compliance - the most compliant actuator undergoes greater displacement. Using the same graphic conventions previously introduced, the relations between mechanical variables is captured by the graph of figure 7. Equation 60 can be derived by following the dashed-line path. Again, the elastic properties of the actuators provide complete connectivity between the mechanical variables of the system. 3. THE HUMAN ARM 1 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ x * x r \ Figure 6. Serial Connection: Actuator Displacement Space. Dashed lines: solution planes for three values of the end effector displacement. Solid lines: solution trajectories for spring-like actuators with known compliance.

186 F.A. Mussa Ivaldi ACTUATORS END-EFFECTOR SPACE SPACE Figure 7. Linear Graph Representation of the Serial Connection. 3.1. Motor Redundancy The redundancy of a mechanical system is defined as the disparity between the number ol degrees of freedom at the end effector and at the actuators. This is a purely kinematic definition, which is subject to significant modifications when dynamics is taken into account [Hogan 1965]. In fact, when the arm interacts with the environment, not only must the kinematic parameters ol the hand (e.g. position and velocity) be specified by the CNS, but also the motor impedance has to be controlled in order to obtain predictable behavior in the face of interaction forces. For instance, the position and orientation of the hand In space Is defined by 6 variables. The rotational degrees of freedom of the arm (the number of independently controlled axes of rotations al the shoulder, elbow and wrist joints) are at least seven. Each of these axes is controlled by several muscles spanning one or two joints. At a first approximation, there are at least 24 muscles involved in the movements of

the arm. The redundancy, expressed as the difference between hand degrees of freedom (dof's) and number of muscles is then: 24 - 6 - 18 , clearly a fairly large number. Such an analysis, however, does not take into account the fact that in real life we are interested In controlling not only the position of the hand but also the response to externa forces by specifying the hand impedance. The latter is expressed, In an initial approximation, by three matrices : the inertia, the viscosity and the stiffness. Each of these quantities is a 6x6 matrix relating, respectively, the acceleration, the velocity and the displacement vectors at the hand to the applied force vectors. Even assuming that these matrices are symmetric, there is e total of 21 x 3 « 63 independent parameters to be controlled beside the 6 positional dof's of the hand.

Compliance 187 Since the term "redundancy" suggests the idea that the number of actuators exceeds the number of controlled variables, its use seems equivocal in a context where the potentially controlled mechanical parameters may even exceed the number of muscles. Below we refer to redundancy not to suggest the existence of an excess of actuators but simply to describe an excess of kinematic variables in the actuator space with respect to the end effector space. In section 2, by means of simple examples, it has been shown that different kinds of connectivity (serial vs. parallel) generate different kinds of redundancy. In a parallel arrangement of actuators, the mapping between end effector displacements and actuator displacements is uniquely defined while, for the forces, the mapping is only unique the other way around. The inverse is true for a serial arrangement: given the end effector force, the actuator forces can always be uniquely computed but given the end effector displacement there are infinite solutions for the actuator displacements if no extra constraint is added. This is the well known problem of inverse kinematics which consists, for a multi articulated arm, in finding out the angular displacements at the joints when the hand displacement is known in terms, for example, of Cartesian coordinates. In the case of the human arm the mapping between the hand and the muscles can be carried out in two steps. The muscles, which are under the direct influence of the neural activity, act around the joints and the connection among muscles and joints is in parallel. In contrast, the joints are serially connected in a kinematic chain to determine the hand position in space. The transformation graph at figure 6 represents these two levels of connectivity. Two aspects are briefly discussed here, which constitute the main differences with respect to the simple cases examined earlier. 1) Force and position variables have different physical dimensions in the three spaces: for example, the displacement vectors in the joint space are angular displacements while in the hand space they are linear displacements. Consequently the force vectors are expressed in terms of torque components in the joint space, while they are ordinary forces in the hand space. As in the Lagrangian formulation of mechanics, forces and positions are generalized concepts which may be "translated" into different kinds of variables according to the particular space in which they are considered. These transformations are expressed by the Figure 8. Linear graph representation of the connectivities in the human arm. Km: muscle stiffness ; Cj: joint compliance ; Kh: hand stiffness.

188 F.A. Mussa Ivaldi Jacoblan matrices relating the different spaces to each other. 2) The Jacobian matrices are, in the case of the human arm, position dependent. This is a consequence of the nonlinearity of both the geometrical transformations, from muscles to joints and from joints to hand The torques generated by the muscles on the joints are given by products of the muscle tensions and the muscle moment arms. The elements of the muscle/joint Jacobian matrix are therefore nothing more than the moment arms of each muscle around each joint which, In general, depend on joint angles. Also the transformation from joints to hand, where trigonometric functions are involved, is non- linear. In spite of these complications, the general conclusions of section 2 remain valid. In particular, the spring-like properties of the actuators give a complete connectivity between force and position variables across the three spaces. The connectivity is provided by the muscle stiffness matrix, by the joint compliance matrix, and by the hand stiffness matrix. While the interactions of the arm with the external world usually take place at the far right of figure 8, the control processes implemented by the CNS operate directly on the right side through neural activation of the muscles. Furthermore, other geometrical spaces need to be added to represent the fact that the motoneuron activity is directed to groups of muscle fibers intermingled within each muscle. In order to approach the complexities of neural control in such a multidimensional domain we may, first, take advantage of the modularity of the motor system. This means, for Instance, that the pathway from the planning of a motor task, in end -effector coordinates, to the generation of appropriate neuromuscular activity can be divided into intermediate steps, such as the description of motor coordination in terms of joint coordinates. Furthermore, the design of experiments aimed at mutti-joint movement control may take

advantage of some constraints which simplify the kinematic redundancy problems. For example, in a first analysis, hand kinematics may be restricted to planar hand movements with two degrees of freedom at the joints (I.e. with the wrist immobilized). This constraint not only symplifies the problems of measurement and control of the experiments but also establishes a bi-directional mapping between hand and joint kinematics and allows the inversion of the corresponding Jacobian matrix in the greatest part of the workspace. Moreover, the muscles may be grouped into three separate sets, according to their action upon the joints. These are: the muscles acting only on the shoulder (single-joint shoulder muscles), those acting only on the elbow (single - joint elbow muscles) and those spanning both joints (double-joint muscles). As we shall see, this simplification leads to some interesting conclusions about the neural controller from the experimental analysis of hand behavior. 3.2. Non-Linear Factors in the Control of Muitl-Joint Posture and Movement In single-joint studies, much experimental evidence has been collected demonstrating that the CNS makes explicit use of the elastic properties of the skeletal muscles in order to maintain stable posture of a limb and to control ita movements. Is it legitimate to extend these findings to multi - joint posture and movement? It isn't possible to answer this question without facing the complexity of multi-joint arm geometry.

Compliance 189 As we have already pointed out, In a multi dimensional system position and force variables must be treated as vectors and the stiffness (more generally the impedance) as a matrix. Furthermore, the geometry of the arm is inherently non - linear, a fact which has important implications for motor control. For example, it has been shown [Morasso 1981] that targeted movements of the hand in the plane follow straight-line trajectories. This is an apparently simple observation which is a result of focusing on arm trajectories in terms of hand rather than of joint coordinates. Since the kinematic transformation from joint to hand variables is non-linear, a straight line in the joint space maps into a curved line in the hand space and vice versa. Hence, the observed regularity in the hand paths led to the suggestion that the CNS plans arm movements by specifying a trajectory of the hand in Euclidean space and deriving the corresponding pattern of coordination at the joint level. This suggestion is important because it reveals a choice that the neural planner takes "by default", i.e. when no hand trajectory requirement is specified by the task. The non-linearity of arm geometry has even more drastic effects when the dynamical aspects are taken into account. The problem of deriving the torques that must be generated to drive the hand along a desired path - the inverse dynamic problem - is complicated by the dependence of the inertial terms on arm configuration and by the presence, in the dynamic equations, of non linear functions of joint velocity, known as the centripetal and Coriolis terms. Consequently, linear feedback schemes are not, in general, sufficient to keep the arm along the desired trajectory, since the choice of acceptable feedback gains depends upon the inertial terms [Raibert and Horn 1978]. The fast solution of the inverse dynamic problem has been, for a long time, a central topic of robotics research. The task is that of computing in real time the joint torques required to drive a kinematic chain along the desired path. The most advanced solutions to this problem are based on recursive formulations of the Newton -Euler equations [Luh, Walker and Paul 1980] and of the Lagrange equations [Hollerbach 1980] which govern arm dynamics. These solutions, however, are largely influenced by the limits imposed on computation by currently available computer architectures which are based on the sequential organization of computational processes. In contrast, the idea that biological information processing may take advantage of the parallel organization of the neural networks is well established [Von Neuman 1958]. More "anthropomorphic" computational schemes for the inverse dynamic problem have been proposed [Benati et al. 1980, Lathrop 1983], based on parallel information processing. It should be stressed, however, that, even before considering specific computational schemes, a more basic question may be raised: is it really necessary that the CNS solves the inverse dynamic problem? The idea that torques must be explicitly computed in order to move the arm derives from considering the muscles as pure "force generators". Muscles, however, do have elastic properties! This simple fact has crucial implications for the processes which generate and control the forces required to drive a limb along the desired paths. For example, we already discussed (see section 1.2) the main experimental evidence suggesting that single joint arm movements are generated by shifting the equilibrium position as it results from the interaction of antagonist muscles. The underlying philosophy is that neural commands do not encode torques or positions but

global control laws, through the choice of length tension functions. Can multi-joint arm movements be represented as shifts of the equilibrium positions dictated by the muscles? If this were the case, the inverse dynamic problem would be implicitly solved since joint torques wouid be generated without any explicit torque computation. Furthermore, the same mechanism would also provide dynamic stability of the arm since any external disturbance occurring during the execution of a movement would be counteracted by the elastic properties of the muscles. As a consequence, the hand would tend to spring back to the original trajectory, even before the onset of reflex responses. In a recent investigation [Mc Keon et al. 1984] subjects were asked to execute planar hand movements between two targets while holding a two - link manipulandum. As the movement was initiated, a constraint was suddenly imposed on the manipulandum in order to deviate the hand from the intended path. The constraint was then disengaged after 200-800 ms and it was observed that the hand Immediately moved back to the original trajectory. Two conclusions were derived from this experiment. First, the reaction upon release of the constraint was too rapid to be explained in terms of reflex responses. Hence it was interpreted as resulting from the elastic properties of the muscles. Second, after release the hand moved not toward the final target but toward what would have been the "unperturbed" trajectory. Thus we can extend to multi-joint movements the hypothesis that they are generated by gradually shifting the equilibrium position of the hand. Three different types of trajectory come into play in such a model for arm movements: the nominal trajectory, the virtual trajectory and the actual trajectory. The nominal trajectory represents the desired movement of the hand. The virtual trajectory is the trajectory of the equilibrium point, as defined by the elastic properties of the muscles and encoded by the neural commands at the peripheral level. The trajectory actually followed by the hand is a result of the interaction of the elastic forces generated by the virtual trajectory with the non linear inertial and viscous forces which come into play during a movement. The task of the motor controller is to reduce the gap between nominal and actual trajectories as much as possible. To this end, one possibility is that of compensating for the dynamic interactions by varying the virtual trajectory. A simple example illustrates this point. Suppose that we want to move a load along a straight line on a flat horizontal surface. The load is attached to one end of a rubber band and we control the position of the other end. The rubber -band defines an equilibrium position (virtual position) for the load, which is the position where no elastic force is generated. At rest , virtual and actual positions coincide. If the load is a simple mass (e.g. a small spherical object), we can drive it by drawing the rubber band along the desired path; the load will simply lag behind the instantaneous equilibrium position. If, on the other hand, the load is a rigid body or a chain of rigid bodies, it will wiggle around the equilibrium trajectory. In order to keep it on a straight path, we have to draw the rubber band along a more complex trajectory. An alternative solution consists of using high stiffness. This would give strong restoring forces even for small deviations from the virtual path. High stiffness reduces the difference between nominal, virtual and actual trajectories. However, high stiffness also generates large forces in the impact with obstacles and creates stability problems.

Compliance 191 From this discussion it follows that the stiffness associated with each equilibrium position of the hand is a crucial parameter in the control of multi -joint movements. Hogan (1960) suggested that for planar arm movements with two rotational degrees of freedom (shoulder and elbow), the double joint arm muscles, such as the biceps and the long head of the triceps, can provide complete control over the three terms of the joint stiffness matrix and, consequently, over hand stiffness. Whether or not the CNS actually takes advantage of such a possibility, in the control of posture and movement, is entirely a matter of experimental investigation. In the following paragraphs the results of some recent experiments addressing this question are are discussed. They represent a first attempt to measure the geometrical properties of the stiffness tensor during multi - joint arm posture. 4. AN EXPERIMENTAL APPROACH 4.1. The Measurement of Hand Stiffness When we keep the hand in a given position, the muscles of the arm act together to maintain stability. This means that, if the hand is displaced by an external force, the muscles generate a net reaction force which tends to restore the original posture. The stiffness matrix associated with hand posture was recently measured [Mussa Ivaldi et al. 1985], The experimental procedure consisted of applying a sequence of position controlled displacements to the hand of the subjects who were asked to maintain a constant posture in the workspace. The restoring forces generated by the subjects before the occurrence of any voluntary reaction were measured during the holding phase of each displacement, i.e. when the hand was at rest. In this condition, the observed forces had no inertial or viscous components. By taking only small displacements it was possible to assume a linear relationship between force and displacement vectors: (6a) F = KSr The next step consisted of estimating K through the repeated measure of F and 6t ■ The main advantage of such a "static" measure is that it is totally independent of any modeling assumption about the differential equations governing arm dynamics. The apparatus sketched in figure (9-a) was used to perform the measurements. It consists of a two-link planar manipulator equipped with two torque motors controlling each of its joints independently. The position of the handle in the horizontal plane was measured by high resolution potentiometers mounted on the axes of the manipulator joints. Subjects were asked to hold the handle in a fixed position while their arm was supported in the horizontal plane, figure (9 - b) is a sketch of the subject's arm under these experimental conditions. The wrist was braced in order to obtain a one to one correspondence between hand position coordinates and shoulder and elbow angles:

 r>Cx,u3 Figure 9. Experimental setup for measuring hand stiffness during planar posture, a) Apparstus b) scheme of subject's arm (top view).

 (63) X — l\- COS0J + li ■ COs(0i + 02) y = h- sin 0i + h ■ sin(0i + 92) where I, and U are, respectively, the lengths of the upper-arm and of the forearm. The corresponding relation for the displacement and rotation vectors is: (84) with Sr = J • 66 J = -(li ■ sin 0, + l2 ■ sin(0, + 02)) -l2 • sin(0i + 02) /[• COS6l + li ■ COS(01 + 62) h ■ COS(0, + 02) The stiffness matrix, defined by (62), has four components (65) K = They were estimated by applying a least squares regression algorithm directly to the two equations (66) {Fz — Jfx.x • Fv = Kv* ■ Sx + Kz<y ■ Sy Sx + K,lV ■ Sy The measured force vectors represent the observed steady state responses to the displacements which were applied along several directions and which had an amplitude ranging from 5 to 8 mm. These forces may contain a contribution from reflex activities induced by the displacements. In fact, when the force was estimated, much care was taken to avoid voluntary responses but no attempt was made to avoid reflex activities. This choice was consistent with a modular approach to motor control since peripheral feedback was considered an integral part of the apparatus controlled by the descending neural activity. 4.2. Coordinate Transformations Once the hand stiffness matrix K was derived, transformation (16) was used to compute the joint stiffness metrix R which relates the joint torque vector to the joint rotation vector: (67) T = R • SO with R = #2,1 fij.j

 Since (68) it follows that T = JT • F (69) R = JT K . J

The knowledge of the joint stiffness matrix represents a step towards understanding of the organization of muscle activities. A further step is the partition of the muscles involved in the control of planar arm posture into three groups of antagonist pairs (figure 10-a): the single joint muscles acting around the shoulder (1 and 3), those acting around the elbow (5 and 6) and those spanning both joints (3 and 4). Since the connection between muscles and joints is of the parallel type, the transformation between muscle forces and joint torques is well defined and can be carried out by utilizing the SMMUCIt : kii kit :ki> : Ul Ktt kit kii' Ktl kii Ml K4I: K«) Ktl KH Ktl KH Ktl Ktl KM U« km: K4«: Kt4 KII Kit Kit KM KM KM Mt K4» Ktt Ktf M4 KM KM Kll Ktl Kll Ml Ktl Ktl ♦mi mt«— Kl) KM Kit Kit Ktl KM Ktf Ktf Kll KM JKM Kit K«) K«« 'Ml K4I KH Ktl Kt« Kit Kll KM Ktf Ktt CM Ktt -»KI RU«- B Kll Kit Ktl Ktt Kll Ktl Kll KM i Kll Ml K4t Ktl KH Ml KM Ktl KH KH KM KM KM; M<: KM KM Kll Ktf KM Mf Kit km Kit at KM MS Kit at Kll ai K>1 Ml Ktl Ul Kit KM KM Mt KH KU Kl) Ktl I til Ml Kl) Ktl KM 114 Kll Kit Kit Kit 04 :nt KM M4 ;K4f K4t KM: Ktl Ktt: If] KM km: Figure 10. Contributions of the muscles to the joint stiffness matrix. A) Partition of the arm muscles according to their action on the joints: 1,2 single-joint shoulder muscles; 3,4 double joint muscles; 5,6 single joint elbow muscles. B)Contributions of the muscle - stiffness terms to the joint-stiffness matrix.

Compliance 195 matrix of the moment arms: (70) T = f • f where f is the 6-dimensional vector of forces generated by the three muscles pairs defined above. The relation between the 6 x 6 muscle stiffness matrix and the 2 x 2 joint stiffness matrix R is then: (71) R = /iT • k ■ n The element k. of the muscle stiffness matrix k represents the change In tension generated on muscle / as a consequence of a change in length of muscle /'. Clearly, since moment arms are also functions of the joint angles, transformation (71) derives from a linearization of the muscle/joint relationship and should be considered as a good approximation only for small displacements. Without going into further details, a few considerations on (71) will be added here. If the muscles were only passive spring-like elements, the matrix k would be diagonal: 0 if x±j -{ As a consequence, both R and K would be symmetric7: (73) R = Rl K = KT However, since reflexes are included in the stiffness measurement, non-zero cross coupling terms between the muscles must be considered and the matrix k is no longer diagonal. Hence, a condition for the symmetry of K and R, whose physical meaning will be discussed later, is (74) i.e. (75) 7 This lotlows Irom

Thus Implies K = K »#y. «,y = i,2,...,6 (atba)t = atbta B = BT (a'ba)' A*BA

196 F.A. Mussa Ivaldi This condition can be phrased as follows: a stretch of muscle / induces, by. reflex activation, a change in tension on muscle / equal to the change in tension on muscle /' induced by a stretch of the same amplitude of muscle /. The contributions of the individual muscle stiffness components to the terms of the joint stiffness matrix are illustrated in figure 10- b. The transformation (71) can be written in a more explicit format as: R{,m = Ei,; Mi,/ • /Vj ' «i,J (76) l,m= 1,2 ; 1 <i,j <6 Here, the (i,i) term of the matrix denotes the moment arm of muscle / with respect to the joint /. Some of the terms in this matrix are equal to zero since the single joint shoulder muscles don't generate any torque on the elbow and the single joint elbow muscles don't generate any torque on the shoulder. This is mathematically expressed as follows: (77) / W.i = 0 if * > 5 1 /ih., = 0 if h < 2 The portions of the muscle stiffness matrices enclosed within solid lines in figure 10-b include all the terms which contribute to each element of the joint stiffness matrix R. It is evident from this sketch that, without any further assumption, the off-diagonal stiffness terms R£ and R, , receive contributions from both the double and the single joint muscles. More precisely, each single-joint stiffness term is influenced by the double joint muscles and by the single joint muscles acting around that joint. In contrast, If the cross coupling reflexes between muscles belonging to different groups (e.g. between single joint shoulder and double joint muscles) could be neglected, then the matrix k would be block diagonal. As a consequence (see the terms enclosed within dotted lines in figure 9-a) only the double joint muscles would play a role in the off diagonal stiffness terms. However the single joint stiffness terms would still be under the influence of both single joint and double joint muscles. 4.3. General Definition of Spring - Like Behavior The most common idea about a spring - like system Is that of a system following the linear Hooke's law (1). This, however, is a restrictive definition, which hardly fits the actual behavior of individual muscles in vivo. A more general definition of a spring-like system is that of a system characterized by a static and Integrable relationship between force and position [Paynter 1961, Rosenberg and Karnop 1983, Hogan 1985). The term static is used to mean that the function describes the system behavior only at zero velocity and acceleration. For such a system a scalar potential energy function U(r) is defined and the force F at each position r is given by

Compliance 197 (78) F = grad (C/(r)) Some consequences of this definition for biological systems, like the human limbs, have been discussed by Hogan [1985]. It has been stressed that individual muscles, when considered as spring-like elements, contribute to the total potential energy of the joints by adding up their potential energy functions. As a consequence, the torque generated about the joints by all the muscles is derived simply by taking the gradient of the total potential energy (which is invariant for coordinate transformations) in the joint space. This property holds independently of the non - linearities in the length - tension curves. During multi-joint arm posture, a field of restoring forces is associated with each hand position in the workspace. These fields have been observed experimentally in several positions by applying servoed displacements, as described in the previous paragraphs (figure 11). In order to state that hand posture has a spring-like behavior, however, it is still necessary to show that the force vectors are integrable functions of hand position. In elementary physics two equivalent necessary and sufficient conditions are given for the integrability of a force field F(r): <79) ^Fdr = 0 (the mechanical work along a closed path, c, must be zero) or, equivalents, (80> cuW(F)^-^ = 0 ox ay It is easy to recognize, in the latter condition, the two-off diagonal terms of hand stiffness BF, By K —ill. (81) ^ y,i — BF, Bx It follows that we may say that hand posture is characterized by a spring-like behavior if and only if the hand stiffness K is symmetric: (82) K = KT There is no a priori reason to assume that condition (82) applies, taking into account that cross coupling reflexes between different muscles are present. (We have shown, in fact, that they might cause asymmetry in the stiffness matrix.) However, in our experimental study it has been found that the force field is conservative. Once the stiffness matrix was derived, it was decomposed into its symmetric and antisymmetric components Kg and K, (83) K = Ks + KA

198 F.A. Mussa Ivaldi l_. I I I U 1 N % * f e » t *• % * \ Figure 11. Force field associated with planar hand posturs. The measured force vectors are represented in a pin-cushon diagram, with the tip on the displaced position. E: hand equilibrium position. (Modified from Mussa Ivaldi et al, 1985).

Compliance 199 using the following formulas: (84) Kg i-(K+KT) * • (K - KT) The matrix Kg expresses the conservative component of the force field which can be derived from a scalar potential function U(r) given, near the equilibrium position r , by: (85) with U=--6rT-KB-6r Sr = r-Te The antisymmetric component KA represents the "rotational" portion of the force field which cannot be reduced to a potential function and which could only be generated by an unbalance between cross coupling reflexes. The relative influence of the two matrices has been established by comparing the forces that they produce in response to several displacements (figure 13). It was concluded, also on the basis of a statistical analysis, that the rotational component KA is negligible and that the stiffness K can be replaced by Kg without loss of significant information. This finding rigorously demonstrates that reflexes, as they are involved in normal posture, do not alter the spring - like nature of the force field associated with the hand. 4.4. The Stiffness Ellipse The stiffness matrix K is a table of four numbers and we have shown that is a symmetric matrix. Therefore, it has only three independent components. However, the numerical representation is not necessarily the best, especially if we are trying to capture their geometrical meaning. Two alternative representations are briefly discussed here: the stiffness ellipse and the iso- potentlal lines. The stiffness ellipse (Mussa Ivaldi et al. 1985) is obtained from the stiffness matrix by plotting the locus of the force vectors when a constant amplitude, rotating displacement is imposed as an input: (86) Kz,z Kx,y *ly,x «y,y cost sin* with 0<< <2-ir The tip of the force vector describes an ellipse (figure 13) which is related to the stiffness

200 F.A. Mussa Ivaldi 2 m in • .« ♦' '.' - 7/ *', ix Figure 12. Comparison between the conservative (a) and rotational components of the force field associated with a measure stiffness matrix. (Modified from Mussa Ivaldi et al., 1985).

Compliance 201 Figure 13. Stifness representations. The same stiffness matrix is represented as a force field (a) or as an ellipse (b). The major and minor axes of the ellipse are proportional, respectively, to the maximum and the minimum stiffness eigenvalues. The orientation is defined as the angle between the maximum axis and the x-axis. The size is the area of the ellipse. (From Mussa Ivaldi et al. 1985).

202 F.A. Mussa Ivaldi parameters as follows: 1) The area of the ellipse is proportional to det(K). 2) The maximum axis is parallel to the maximum eigenvector of K. 3) The ratio between the lengths of the maximum and

minimum axes is equal to the ratio of the two eigenvalues of K. Apart from any rigorous definition, the meaning of the stiffness ellipse is straightforward: its orientation is that of maximum resistance (maximum stiffness), its siza is related to the "total" amount of stiffness along all directions and its shape, more or less elongated, reflects the degree of Isotropy of the restoring force field. It is important to note here that the restoring force is not always oriented along the line of displacement. Force and displacement are in fact collinear only on the two axes of the ellipse. If the off-diagonal terms of K are zero (figure 14 a and b), then the ellipse is oriented 1) along the x-axis, if Kxx > K , and 2) along the y- axis, if Kx x < K . When K « K and K « K « 0 the ellipse degenerates into a circle. In this case all x,x y,y x,y y,x r ™ the directions are eigenvectors of K and the force is always collinear with the displacement. Such a condition Is also referred to as "isotropic" stiffness. In figure 14-c the ellipse of figure 14-a has been rotated 30 degrees. The corresponding matrix has non-zero off-diagonal terms. The three stiffnesses of figure 14 have the same size and shape. They only differ in orientation. This is the kind of Information which is made evident in the geometric representation of the stiffness and which may be concealed in its numerical expression. An alternative representation for the stiffness is provided by the potential energy function [Hogan 1980]. Eq. 85 defines, for each value of U, an elliptical contour (isopotentlal line) in the displacement space. If a third axis - U - is added to the two axes a surface is obtained in this 3-D space for each value of K (figure 15). Vertical sections through the origin of this surface are parabolas and horizontal Isopotentlal sections are ellipses. The shape of this surface is that of a valley, or a bowl, whose lowest point is at the equilibrium position of the hand. For a given stiffness matrix, the isopotential ellipses are orthogonal to the corresponding stiffness ellipse. Furthermore, the shape factor (i.e. the ratio of the maximum and minimum axes) of the isopotentlal ellipse Is the square root of the shape factor associated with the stiffness ellipse: (87) ll^MAxll _ I Km AX II^minII V Km,n Here, H^rw^ll and Il^r|\/t||sjll denote the maximum and the minimum displacements from the equilibrium, at a given potential energy U, while K»,,x and KM,N are the maximum and minimum eigenvalues of the stiffness matrix. Hence, the stiffness ellipse representation is more sensitive to the difference between KMAX end K>..N and, therefore, is a better candidate for measurement purposes. However, the potential energy ellipses have the advantage of providing an immediate "visual" Insight about the direction of the restoring force corresponding to any

Compliance 203 •8»o e -tee e -17S -43 -ae«J / Figure 14. Relationship between stiffness representations: numericel (left) and ellipse (center). Right: corresponding force vectors along the x and y axes. The stiffnesses in a, b and c have the same shape and size but different orientations. Note that the forces are parallel to the displacements only along the principal axes (a and b).

204 F.A. Mussa Ivaldi Figure 15. Three perspective views of a potential energy function. The elliptical lines denote constant pi tential energy (U) levels (isopotentlai lines). The parabolae passing through the origin are orthogonal to tti isopotantlal ellipses and therefore represent gradient lines. given displacement. In fact, the elastic force is always orthogonal to the isopotential line passing through Its point of application. Furthermore, as we already pointed out, (78) holds Indepen dently of the particular coordinate system which Is considered. Both the potential energy curves and the stiffness ellipses can be generalized to represen other components of the motor impedance. For example Asada [1983] uses isokynetic energy ellipsoids to describe the inertia associated with the end effector of a robot arm. This led recently to a new method for designing artificial manipulators from the prescription of a desired inertia behavior in the workspace [Lee 1984], An alternative representation for the Inertia consists o plotting the locus of forces (inertia ellipse) generated by the equation (88) F = M u(«) where (89) u(0 = [cos(0,sin(0]r 0<*<2ir M = JTIJ Here, I is the inertia matrix of the manipulator, relating joint torques to angular accelerations, am J is the Jacobian which transforms joint velocities into end effector velocities.

Compliance 205 4.5. Independent vs. Synergistic Control of Muscle Stiffness We have shown that, in theory, there is the potential for the CNS to modulate the hand stiffness by modifying the individual elements of the muscle stiffness matrix (71). This observation, together with the experimental evidence that muscles do change their stiffness when the alpha - input varies [Rack and Westbury 1969, Agarwai and Gottlieb 1977, Hotter and Andreaeen 1981] led to the hypothesis that the CNS might modify the geometrical parameters of the elastic field at the hand by modulating the individual joint stiffness terms according to the requirements of the task [Hogan 1980]. On the contrary, the experimental results obtained by Mussa Ivaldi et al. [1985] do not support this idea. First, a consistent and repeatable pattern of hand stiffness has been found when different subjects were asked to maintain hand posture in several locations of the workspace (figure 16). Furthermore, when the subjects were asked to maintain hand posture against a predictable force acting along a constant direction, the orientation and the shape of the stiffness did not change Figure 16. Variations of hand stiffness in the workspace measured in four subjects. The segments represent the upper arm and the forearm at each location. S: shoulder E: elbow. (From Mussa Ivaldi et al. 1985.)

206 F.A. Mussa Ivaldi substantially with respect to the normal postural stiffness in the same position. In contrast the size was found to Increase by more than 100 %. These facts suggest that changes of muscle stiffness are uniformly distributed over the arm muscles. In fact, if all the elements of k in (71) were scaled by the same factor, then also the joint stiffness and the hand stiffness terms would scale uniformly by that factor. This would result in a change of stiffness size without any change of shape and orientation, as was actually observed. These findings then suggest a synergical organization of muscle stiffness. Here, the concept of synergy indicates that, in spite of the redundancy in the neuromuscular system, the mechanical behavior of the hand is not subject to the expected variability. Hence, we have found a constraint which, ultimately, must be satisfied by the distribution of neural activity among the muscles. Whether this constraint is satisfied by the topological organization of the motoneurons and of the reflex pathways or by more central processes is still an open question. A similar approach, involving the deduction of neural synergies from the analysis of biomechanical constraints, has been proposed for the analysis of human standing posture by Nashner and McCol- lum [1985]. Another constraint was found in the pattern of hand stiffness obtained by asking the subjects to maintain different positions in the workspace. This pattern may be defined as "polar" since the ellipses were approximately oriented with the major axis pointing toward the shoulder. An ideal polar manipulator, which would give a similar pattern of end - point stiffness, is depicted in figure 17. It consists of two joints: one is a revolute joint centered in the origin of the reference frame (the "shoulder") and the other is a prismatic joint (the "elbow") which allows sliding of the distal link with respect to the proximal one. The reason why such a manipulator is called polar Is that two actuators, independently controlling the position of each joint, would control the end point position in terms of polar coordinates: (90) fx = ,.cosJ If the actuators were spring - like, the resulting end point stiffness would be oriented along either one of the looal axes x' and y' which represent the directions controlled

independently by the actuators. In spite of the radical difference between such a manipulator and the human arm, the stiffness measured on human subjects followed a similar pattern. In a subsequent work, Flash and Mussa Ivaldi [1984] expressed this observed behavior of hand stiffness as a condition which must be satisfied at the joint level. In fact, it was demonstrated that a necessary and sufficient condition for hand stiffness to be polar is that the shoulder joint stiffness term Is exactly twice the double joint stiffness term: (91) #i,i — 2 ■ Ri,2

Compliance 207 V ' V \ / v v / /Pa VSft ^ *^"l Figure 17. Two-link polar manipulator. without any requirement for the elbow stiffness.8 This equation may be derived by making the stiffness matrix diagonal when expressed in polar coordinates (a condition which is equivalent to the requirement that the polar actuators of figure 17 are independently controlled). Then, the joint stiffness matrix is derived from (92) R = JJV Kp Jj. where K_ is the stiffness matrix in polar coordinates, and J is the Jacobian which maps joint displacements into polar displacements: (93) s<t> l-p ' [69, Eq. 91 has been found to fit the actual joint stiffness data computed (eq. 69) from the measured hand stiffness in several workspace locations: during planar posture the shoulder and the double joint stiffness terms followed the same trend as the shoulder and elbow angles changed (figure 18 a-b). In contrast, the elbow stiffness (figure 16-c) followed an independent pattern of variations. These results suggested the existence in the workspace of a synergistic coupling between double joint muscles and single joint muscles acting upon the shoulder. Whether or not the polar configuration of the hand stiffness simplifies the control of hand behavior, still has to be investigated. However, some observations could be made on the feet 8 This condition holds strictly in the hypothesis that the upper arm and the lorearm heve the same length. In the case ol Ihe human arm, this is a good approximation when the wrist is braced and the lorearm length is computed from the elbow to the center ot the hand.

208 F.A. Mussa Ivaldi smouuoi aTimcsf v* joint ukui MSU-JOIKT (TimCM * JOINT «HSUS Figure 18. Measured variations of the joint stiffness terms in the workspace. Independent variables: joint angles (shoulder and elbow). Dependent variables: stiffness terms (N m / rad). The surfaces have been obtained from the experimental points (squares) by least square rsgression on a 3rd degree polynomial function, (from Flash and Mussa Ivaldi 1985).

Compliance 209 that a polar manipulator gives, at the end effector, a locally orthogonal coordinate system whose axes are independently controlled by the two actuators. In contrast, a manipulator with two independently controlled revolute joints defines a more complex local geometry at the hand. In fact, the two directions, at the end effector, controlled independently by the shoulder and by the elbow actuators form an angle which varies in different workspace locations. In other words, polar geometry lies between Euclidean geometry, which is best suited for movement planning, and the complex articular geometry of the human skeleton. This compromise may be achieved by relatively simple co - activation schemes at the muscle level when cross coupling between the joints is possible. The constancy of the stiffness orientation associated with each arm configuration does not support the hypothesis that the CNS is tuning the individual joint stiffness elements according to task geometry [Saltzman and Kelso 1963, Saltzman 1985]. If this were the case, one would expect an adaptive reorientation of hand stiffness when a subject is given the task of keeping the hend at rest against a perturbation acting in a constant direction. On the contrary, only changes in stiffness size have been observed in these experimental conditions. However, even with fixed distribution of stiffness over the joints, the hand stiffness can be still modulated by taking advantage of the kinematic redundancy of the human arm. Indeed, when the same hand position is reached with different joint angles, different hand stiffnesses are produced by different arm Jacobian matrices. By the same argument, not only may hand stiffness be modulated, but also other components of the impedance such as the inertia and the viscosity. A discussion about the control of the mobility ellipse (i.e. the inverse of the inertia) by means of the redundant degrees of freedom of the hand can be found in [Hogan 1985]. Stiffness, viscosity and inertia play different roles for different frequency contents of the external disturbances. While in steady state the dominant source of reactive forces is the stiffness, the hand behavior is strongly influenced by inertia during the transient state. As a consequence the choice of an optimal posture in the execution of a task depends on the frequency content of the expected interactions with the environment as well as on their directional properties. 5. CONCLUSIONS The purpose of this chapter is to introduce the main geometrical Issues related to the interactions of a multi-joint arm with the external environment. It is also to show that any theoretical approach to this topic must and can be supported by experimental work. The geometrical properties associated with multi - joint compliant behavior come into play at two levels: in the representation of a motor task and in the transformation between the behavior of the hand, where manipulation takes place, and the control signals delivered to the muscles. At the task level it was shown that the impedance, which relates force to position variables, must be treated as a multi dimensional object. In particular, when planar manipulation is considered, the stiffness associated with the hand is a 2 x 2 matrix which can also be represented

 as an ellipse. This pictorial representation provides an immediate Insight into the geometrical aspects which characterize the static response of the hand to external perturbations. In a more general multi-joint situation, the hand stiffness is a 6 x 6 matrix and it seems more complex to find a compact pictorial representetion. However, the translational and rotational components of the stiffness matrix can be separately represented by three - dimensional ellipsoids whose parameters are defined in the same way as in the planar case. The dual character of force and displacement variables has been discussed with respect to the transformation from end point to actuator variables. This duality is also relevant at the level of task description where force and position control play a symmetric role. Starting from this observation, Mason [1979] devised a general framework to create control schemes for robot manipulators interacting with the environment. The task geometry was described as a set of constraint equations involving force and position variables in a mutually exclusive way. From these constraints and from the definition of a goal trajectory a control strategy was defined by Mason as a unique set of artificial constraints which are orthogonal to natural constraints. The artificial constraints establish which variables must be position controlled and which must be force controlled. For example, given the task of inserting a peg into a hole no movement is allowed against the surface of the hole while no reaction force occurs along the direction of insertion. As a consequence, the axes normal to the hole surface are force controlled while the axis of insertion is position controlled. Such a scheme allows for some mismatch between the ideal and the real description of the constraints without jeopardizing the proper execution of the task. However, as pointed out by Hogan [1980], "Position control and force control are simply degenerate or extreme cases of impedance control." A manipulator is characterized, in ideal position control, by infinite stiffness and, in Ideal force control, by infinite compliance (zero stiffness). All the intermediate grades are allowed only if the stiffness, or the impedance, is explicitly controlled. For example, the task of inserting a peg into a hole could be executed by tuning spring-like robot actuators in order to have the major axis of the end point stiffness oriented along the line of insertion. In general, by adopting Mason's formalism, the end effector stiffness ellipsoid should be kept with the major axis parallel to the natural constraint surface. A non-zero stiffness in the direction normal to the constraint surface would result in a constant "bias" force if the end effector equilibrium position penetrates the constraint boundaries. Then, contact forces and sliding forces can be modulated by appropriate planning of the virtual trajectory and of the end point stiffness [Morasso and Mussa Ivaldi 1962b). The other level at which geometry is relevant to the analysis of multi-joint impedance relates to the transformations between actuators and end effector variables. It has been shown, by simple examples, that the connectivity of the actuators is represented by transformation rules between variables expressed in the end effector space and in the actuators space. In the case of the human arm, the muscles are connected in parallel to the skeleton and the skeletal segments are connected together in a serial kinematic chain. Consequently, the mechanical variables may be represented in three spaces: the muscle space, the joint space and the hand space. The

 transformations between these spaces are redundant and non-linear. The non-linearity determines the position dependence of the Jacobian matrix, while redundancy can be expressed in two different ways, both relevant to motor control. The redundancy of the skeletal system produces a non-unique mapping from hand to joint displacements while the redundancy of the muscular system is expressed by a non unique mapping fron joint torques to muscle forces. However, it has been shown that, if muscles are considered as spring-like elements and if the muscle stiffness matrix is given, a unique and well defined mapping is established between forces and displacements m the three spaces. The kinematic redundancy of the human arm is discussed here as a potential factor which can be used by the CNS to modulate the apparent impedance of the hand [Hogan 1985). The classic problem of motor equivalence [Bernstein 1967] indeed derives fron the non uniqueness of the co-ordination pattern corresponding to a given motor task. However, when the dynamics of the interactions with the environment are explicitly taken into account, the task space itself potentially has many more dimensions than the few kinematic degrees of freedom of the hand: as a consequence, many of the co-ordination patterns generating the same hand movement are simply no longer "equivalent". The number of redundant muscles acting on one or two joints also provides the CNS with the possibility of complete control of the hand stiffness matrix. For example, shape size and orientation of the stiffness ellipse associated with planar hand posture might be modified by selective "tuning" of elbow, shoulder and double joint muscles. An experimental approach to this problem, based on the static measure of planar hand stiffness under several postural conditions, has been discussed here. When the experimental data were represented as stiffness ellipses, an invariant pattern of shapes and orientations was found across the workspace. This pattern was preserved even when the overall neuromuscular activity changed in order to compensate for external forces of predictable magnitude and orientation. It was concluded that the CNS distributes the stiffness changes over the muscles in a rather uniform way instead of modulating individual joint components. In fact, while hand stiffness is strongly influenced by the geometry of the joint and of the muscle attachments, these geometrical factors, expressed by the Jacobian matrices, may, by no means, be the only cause of the observed patterns of shape and orientation. For example, it has been shown [Flash and Mussa Ivaldi 1984] that, on the basis of the moment arms alone, a pattern of elbow joint stiffness should be expected which is completely different from that actually observed in multi joint posture. An important - although perhaps obvious - assumption supported through this chapter is that complex behaviors, such as multijoint arm posture and movement, cannot be fully understood in a reductionist approach to the motor system. More specifically, one cannot gain complete knowledge of multi dimensional motor co-ordination by simply adding up the information gathered in single joint studies. An example of the qualitative gap between single and multidimensional domains is the definition itself of a spring-like system. It was demonstrated that the postural force field generated at the hand by the combined action of the muscles is spring -

 like since the stiffness matrix is symmetric. From that finding a constraint was derived which must be satisfied by the cross coupling reflexes which relate the mechanical behavior of muscles spanning different joints. In contrast, on the basis of the observations on single joint behavior, where the interactions between joints are not taken into account, no conclusion can be derived about the integrability of the force displacement relationship at the hand level. The qualitative difference between single and multi- joint results has already been pointed out in the kinematic domain [Morasso 1981, Abend et al. 1962] where complex joint coordination patterns are actually generated in order to obtain simple hand trajectories in Euclidean space. The same kind of geometrical complexity must be explictly faced, both at theoretical and at experimental levels, in order to understand how the Central Nervous System implements the, as yet unparalleled, grace and dexterity of human manipulation.

PART II Chapter 2 COGNITIVE MODELING OF PURPOSIVE ACTIONS G. Adomi, S. Gaglio, L. Massone, P. Morasso, R. Zaccaria CONTENTS

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasoo (Editors)
COGNITIVE MODELING OF PURPOSIVE ACTIONS G. Adorni, S. Gaglfo, L. Massone, P. Morasso, R. Zaccaria Planning, World Modeling, and Skill are the three main components of an Intelligent actor, without any rigid hierarchy among them. As a sort of coordinates In an abstract knowledge space, these aspects appear at different levels of analysis of purposive behavior. The particular type of knowledge which is distributed across the different layers is discussed, from geometric knowledge to qualitative physics, emphasizing the complementarity of prepositional vs. analogical representations. The notion of Open System is taken into account showing how it captures much of the complexities of purposive behavior. Two specific paradigms are analyzed In more detail: the Assembly Paradigm and the Navigation Paradigm. 1. PURPOSIVE ACTIONS A purposive action is a coordinated movement of an

actor in a working environment performed in order to accomplish a goal. Therefore, in modeling purposive actions we must consider - at least - three issues: 1. Methods for generating coordinated movements of an actor {skills). 2. Techniques for modeling a working environment {geometric and functional modeilnfj) and for representing the common sense knowledge about the physical world (qualitative physics). 3. Methods for accomplishing a goal {planning). The first topio refers to what people can do1, the range of performances that can be expected 2, the conditions of applicability and the coherence of co - ordinated movements 3, and the composition of skills In terms of motor primitives. The different aspects of this topic are addressed in Part I of the book. ' Humans can "walk", "run" or "pickup- but cannot "fly" or "cut" unless they use special machines or tools. It is Impossible lor a human or humanoW robot to pickup something that weighs 1000 Kg, to walk at BO Kmph, to pan through a door 1omx2cm, or to hold ten balls. 3 K ia impossible to walk and run at the same time but it is possible to move the hand along a trajectory on a surface and to push on the surface at the same time, as In painting with a paint brush.

216 G. Adorni et al. The second topic refers to the techniques for modeling the shape of objects and scenes not only from an objective point of view - as in CAD/CAM - but also from a subjective point of view. The actor, in fact, is likely to have available only a rough description of the environment at the beginning of the action and must be able to update it incrementally during the action, as a result of his subjective, local experience. These issues are addressed in Part III. Moreover, world modeling is clearly related to common sense knowledge about the physical world, an issue addressed by naive and-or qualitative physics [Bobrow 1983]. The third topic refers to what is known (in the Al field) as planning, that Is the techniques for finding the best - or simply a good - sequence of actions which are needed for obtaining a goal from an initial state 4. Planning is a process that precedes the actual performance. However, there are various ways In which planning may condition acting: it may inexorably "guide" it, step by step, or simply "inspire" it, leaving many details to be filled up at run-time, i.e. during the actual performance. In any case, planning implies an Interface between subjective knowledge and objective knowledge. Knowledge, intelligence, and competence need to be distributed among the three main aspects outlined above - planning, world modeling, and skill. In other words, our modeling approach goes against any

hierarchical formulation of intelligent and purposive behavior. Moreover, it is important to compare the above classification of purposive behavior with the more traditional one - perception, movement, and cognition. From the point of view of generality, the two classifications are equivalent because either skills, or world modeling, or planning have perceptual - motor - cognitive aspects. The difference is that a perceptual - motor - cognitive formulation Is mainly subjective, actor-centered, whereas a skill-world modeling-planning formulation is more balanced: knowledge about oneself, about the world, and about their interactions. This formulation is not only meaningful at the macroscopic level of global behavior but also at intermediate levels, except for microscopic phenomena of sensory or muscular kind. We deem indeed that a sort of sell similarity 5 characterizes purposive behavior: When we analyze actions with ever increasing deteil we find that behavioral units preserve the same type of integrated acting - modeling - planning. This is obviously related to the distributed nature of the central nervous system architecture, where the localization of functions In the brain is very rough whereas sensory-motor-cognitive aspects are present everywhere, although with a varying degree and type of mix. This kind of arrangement is partly captured by what we may call the "orchestra metaphor". Each member of the orchestra Is carrying out a specific musical task but all members also share a global knowledge about the planned performance of the orchestra and a general knowledge about music and performing. Therefore, the same kind of knowledge is instanced in every member and this knowledge is perceptual, motor, and cognitive at the same time. Although this 4 A "stale" is a parametrized description ot the working environment and ol the actor(s). The richness/complexity ot the "stale space" is a lunction ot the level ol detail lhat we wish to make explicit. 5 Sell - similenty is a concept investigated in the framawork of Fractal Geometry |Mandelbrot 1983). Fractal shapes heve inllmte degreas ol details which can ba revealed by increasing levels ot 'magnification*. Self-similarity characterizes the lact that the "structure* revealed at increasing magnification remains invariant.

Cognitive Modeling of Purposive Actions 217 arrangement is highly redundant, it is extremely flexible and robust because no collapse of the performance of the ensemble is caused by the absence of any member, including the director. 2. REPRESENTING SKILLS A skill, as discussed in Part I, is a formal description of a motor activity which emphasizes selected aspects of the interaction with the environment and other actors in a tunable, programmable way. A skill implies a distorted representation of reality, in a similar way to the locality of the human visual system, which has a non-uniform resolution over the field of view [Massone et al. 1985] and aims the peak - resolution part - the fovea - to the area where attention is needed. Similarly, in a skill attention may be focused on certain geometric, kinematic, dynamic entities or specific events, disregarding everything else as "blurred periphery" to be taken into account only in a rough way, e.g. as an emergency signal. A skill can be represented as the concurrent interaction of a number of actors 6 and several skills can concurrently co-operate in purposive behavior. An actor can be recursively defined; a joint, a limb, a man, a team are all actors, possibly resulting from the co-operation of other actors. It can also be an abstract computational activity - a simulation of physical processes - ss well as a controller devoted to co - ordinate the activity of other actors. A skill is an abstract and generic representation able to adapt itself to different contexts (walking at different speeds, climbing, avoiding obstacles, going upstairs, etc.). Therefore, skills are integrated perceptual-motor-cognitive units. A skill is not only a mechanism to generate actions but also a method of representing knowledge about actions in an analog way. This kind of knowledge may be alternative and/or co-operative with the kind of propositions! knowledge, which is related to qualitative reasoning. Consider, for example, the action of putting down an object. The necessary knowledge can be embedded in the skill (formal description adaptable to different contexts) or can be expressed explicitly as a set of qualitative attributes (planarity and suitable shape of the supporting surface, mechanical robustness, speed of approach, etc.). In this case, the propositional qualitative representation can be integrated with the skill: it constrains the skill's semantics making it more suitable for propositional computational processes typical of planning activities. A skill can be used, as already pointed out, as a simulation tool; as a consequence, its qualitative properties can be derived when necessary. A skill, as a unit of behavior, is self-standing to a high degree: It can sense the environmental situations, it can check validity conditions for its application, it can estimate and signal the changes it generates. In other words, any skill reproduces, at a smaller scale, the same qualitative and structural features of global purposive behavior. This implies a "democratic" arrangement in which planning, decision making, adapting etc. are distributed among all the components

218 G. Adorni et al. of the global system. As outlined above, a skill implies - as its components do - both world modeling and planning. A similar conclusion, however, can be reached if we start our analysis with world modeling or planning. The apparent paradox points out the multipolarity of purposive behavior - an instrument which can be played in many different ways. 3. REPRESENTING THE WORKING ENVIRONMENT The real world is a continuum of infinite complexity and common sense knowledge can only hope to approximate it with a network of concepts. The network, as pointed out by Hayes [1979], should be structured and formalized in such a way as to 'cover" some significant piece of physical reality completely, faithfully, densely, and uniformly. Taking for granted that a representation of the physical world can never fit it perfectly, the research problem which is still investigated is to avoid on one side the explosive complexity of the representation and on the other the triviality of a "toy world". In particular, we deem it important to link physical concepts with sensory- motor concepts, aiming at a representation which is common to both of them. The basic tasks of an intelligent robotics system in the real world are manipulation and sensing. These both have physical objects as their basis, and so all computer representations of the real world must have at their foundation a representation of physical objects. The definition of an object in robotics is that set of particular physical properties of concern in sensing and manipulation. Most fundamental are geometry and topology. Of great importance to manipulation are masses, centers of gravity, coefficients of friction and more. Of great important to sensing are reflectivity, color, magnetic properties and so on. Computational representations of objects were studied initially by people in the CAD/CAM field and only recently in Al. Serra [19821 and Requicha [1980], among others, emphasized the importance of a well defined mathematical formulation of the notion of object, in order to derive meaningful computational schemes. Formal and informal properties of a representation can be defined, such as descriptive power, validity, completeness, uniqueness, conciseness, efficacy of the application context etc. Among the different representational schemes which have been investigated, we may quote the following ones: pure primitive Instancing, spatial occupancy enumeration (voxel by voxel), cell decomposition (it Includes the octree representation), medial axis transform, constructive solid geometry, sweeping representation (it includes the generalized cylinders), boundary representation. In general, a suitable geometric representation of the environment is one that distinguishes,

in the shape of objects, between a "main structure" and the overlaying "details" or "textures". For example, different door keys might be characterized by a similar structure (an oblongated protuberance linked to a rounded body) but by different details (the teeth of the keys). We may introduce a complementarity principle for the description of a shape (see the first chapter of Part III):

 Shape demands description at different levels of detail: globsl descriptions require low detail whereas local descriptions require higher detail. The relative emphasis depends on the particular purposive action. The complementarity principle implies that the shape of natural objects cannot be described uniquely: it is possible indeed to go from the molecular to the macroscopic structure passing through a large number of significant levels of description. The hypothesis that is investigated in the previously mentioned chapter is that the two entities can be separated in terms of frequency analysis, the low frequency component being associated with the main structure, and the high frequency component with details/textures. It is worth noting that the same complementarity principle, here formulated with respect to Shape Analysis, is also meaningful in Skill Analysis, discriminating between prepositional and analog aspects of action. This principle leads to qualitative reasoning, on one hand, and to analog simulation, on the other. 4. REPRESENTING QUALITATIVE KNOWLEDGE The recent developments in Artificial Intelligence have focused on the central role of Knowledge, as a dynemic body of facts and rules, abandoning for the most part the pursuit of General Problem Solvers, applicable across all domains. The role of logic itself has been somehow downgraded. The power of logical methods lies indeed in their representation of the world in symbols that can be manipulated in well - understood ways to produce inferences. That power, however, is also their greatest weakness. Most real world problems do not lend themselves to representation through precise logical formalism: heretofore these problems have been considered ill - defined and unstructured and thus difficult to computerize. Hewitt has investigated at depth the reasons for this situation and many of the key aspects are captured by his notion of Open System (Kornfeld and Hewitt 1981, Hewitt and De Jong 1983]. 4.1. Open Systems An open system is characterized by perpetual changes of its knowledge, structure, goals and by decentralized decision making which is bssed on "weighing" partial knowledge, beliefs, dogmas etc.. As a consequence, the knowledge bases of an open system are systematically inconsistent - from the logical point of view. In other words, an open system violates the * This is an exteneion of the concept defined by Hewitt et al |1973).

220 G. Adorni et al. closed-world assumption 7 that underlies most part of the work in A.I. and database techniques. Inconsistency is the consequence of the dispersed, asynchronous, incremental nature of human knowledge. From the point of view of logic and truth-theoretic semantics, Inconsistent data bases and knowledge are purely meaningless because they correspond to no possible (closed) world. However, this is not the only possible formulation. If we shift the focus of attention from truth to communication, we may define message - passing semantics - a semantics where the meaning of messages is determined by its effects on the recipients. Therefore, meaning is not unique (the same message may have different effects in different contexts and at different times) and inconsistency is simply an unavoidable side - effect of the communication - based nature of open systems. In a sense, Inconsistencies in knowledge systems are like diseases in biological systems: They are noxious but intrinsically connected to life. However, although diseases cannot be eliminated, they can be managed in different ways: Diseases can be prevented by healthy practices (carefully selected rules and beliefs) and can be cured by means of medicines (selective "inconsistency killers"), surgery (macroscopic removal of pathological subsystems), and transplantation/prosthetics (structural changes of part of a system). Although the open systems theory was conceived for the study of what we may call the "physiology of large-scale organizations" [March and Simon 1963, Barber et al. 1983], - particularly in the application area of office automation [Ellis and Nutt 1980, Barber 1982] - we think that it provides a valuable insight for understanding the nature of human movements and purposive actions. Purposive action is indeed an example of behavior which is systematically fed by partial, dynamic, inconsistent knowledge (about the environment, the body, the goal of the action). When modeling skilled actions, as well as the behavior of an organization, logical reasoning is an important module but it is not the main one and it does not certainly hold the leading role. For formal reasoning to work, the problem domain must be small. One application of formal methods - relevant for human movement research - that may prove fruitful in the near future is the simulation of qualitative physical reasoning. Explaining how physical systems work is indeed an important requirement for intelligent

software dealing with the complexity of the real world. 4.2. Qualitative Reasoning about Physical Phenomena Natural phenomena - mechanic, thermic, etc. - which can be picked up by our senses are dominated by complex dynamic constraints (ordinary or partial differential equations) which in many cases are too complex to be managed by means of the standard mathematical methods. Yet, humans are quite able to capture - in most cases - the qualitative behavior of a system or 7 The closed-world assumption Implies, lor exampis, that It It Is not possible to prove the truth ol a statement, then the statement must be lalse. The opposite situation is represented by the resolution- by -refutation {reductio-ad- absurtuirl). which consists ol trying to establish the truth ol a given statement by demonstrating thai negating thai statement leads to a contradiction: II no proof is lound, then the statement is considered un-decldible.

Cognitive Modeling of Purposive Actions 221 process which may help deciding, for example, whether or not an object will fall or break and then devising an appropriate course of action. A common criterion used in the research on qualitative reasoning is that the (qualitative) behavior of a system must be derivable from the structure of the system. In order to do that, a qualitative differential calculus is being developed [De Kleer and Brown 1984, Forbus 1984, Kuipers 1984] in which both the time axis and the state variables are quantized in order to deal with continuous systems or mechanisms in symbolic terms. Quantization transforms a quantitative statement (for example, the input-output equation of a mechanical component) into a qualitative one (the following statement, for example: "the increase in input 1 causes a decrease in output 1 and an increase in output 2, whereas a decrease in input 2 causes decreases in both outputs"). A common method of quantization takes only three values (+ - 0) to denote increase, decrease or stationarity (Incremental Qualitative Analysis). The time axis is also quantized, marking qualitative significant events, as a result of qualitative behavioral simulation (also called envislonment). In other words, the qualitative descriptions of variables, in time and space, describe the order of things rather than the things themselves. The qualitative description of a system consists of a large set of qualitative equations (called confluences) which make explicit the ceusal relations among events (e.g. "opening a valve causes a flow increase"). The behavior of a system is then viewed as arising from the interaction of a set of processors, one for each component of a system. The interactions among the individual components are essentially message - passing transactions, each of which may be viewed as a cause-effect relation. Hence, causality has the meaning of message - passing in large physical interconnected systems. From this point of view, there is no difference between the behavior of physical phenomena and the behavior of machines: the physical world is viewed as a machine (albeit a very complex one). Different techniques have been developed for "envisioning" the behavior (or the set of possible behaviors) of a mechanism. These techniques, which have an obvious similarity with message - passing systems or actors systems, do not only provide a support for prediction but also for explanation ("How did a mechanism produce a predicted behavior?") which is the basis of question answering. 4.3. Qualitative Process Theory Let us now discuss in greater detail, in relation with human movement planning, the Qualitative Process Theory developed by Forbus [1984] . Processes are viewed as physical phenomena - heating, stretching, colliding, etc. - that cause changes in objects and the formalism defined to capture them consists of four main elements:

 1. Individual Views. They are used to describe both the contingent existence of objects and object properties that change qualitatively with time. 2. Quantity Spaces, that implement the qualitative representation of physical entities, generalizing the Incremental Qualitative Analysis by De Kleer [1984]. A quantity space is a partial ordering among values: for instance, you may say that a given quantity is greater than a given threshold and Its time derivative is positive at a given instant. 3. Qualitative Proportionality, that describes the functional dependencies between quantities. 4. Histories, that represent what happens to objects over time. Let us assume, to make an example, that we wish to provide the qualitative representation of elastic objects, that Is objects that stretch whenever they are pulled, starting from some initial "rest__length". The formal definition - according to the Forbus formalism - is the following: Individual View Bastic_Object Individuals: B an object Preconditions: Elastic_Substance(made_of(B)) Relations: Has_Quantity(B,length) Has_Quantity(B,internal_force) Has_Quantity(B,rest_length) Direst length(B)]-0 in!ernaFforce(B) alfaQ length(B) Correspondence ((internal force(B) - ZERO) (length(BT-rest_length(B))) An individual Instance of an elastic object becomes "active" when the "preconditions* are fulfilled. Then, the "Relations" part of the definition becomes valid. Specifically, It becomes true that B has particular

types of quantities (length, Internal force, rest length), that the rest length Is stationary s, that the internal force of the object is qualitatively proportional to its length, and that the two relations - "the internal force is zero" and "the object length is equal to the rest length" - are equivalent. Elastic_Object is the basic view of an elastic object relating internal force and length. Other views cen be used to describe the states in which the object can be: Relaxed, Stretched, Compressed. For example, the Stretched_State can be represented by the following view: lndividual_View Stretched Individuals: B an Elastic_Object Quantity Conditions: A~l|ength(B)] > A [rest_length(B)] The changes of objects are modeled by means of Processes, that Is entities that act through time ' The formalism lor dealing wilh quantity spaces consists ol two symbols - A, D - to represent the Amount and the Derivative ol a quantity, respectively. For each ol the two entitles it is possible to reter to the sign - A., Ds - or to the magnitude - Am, Dm. Another symbol - allaQ - is used to denote qualitalivt proportionality, either positiv* or negative according to the sign.

Cognitive Modeling of Purposive Actions 223 to modify the parameters of objects in a situation. In the case of an elastic object, we may define processes to describe Stretching, Compressing, Relaxing, and Breaking 9. For example, the process of Stretching can be represented as follows: Process Stretching Individuals: B an Elastic_Object Preconditions: Position_Constrained(B) is false Quantity_Conditions: Compressed(B) is false A (applied_force<B)] - 1 Am [applied force(B)] > Am [internal_force(B)] Relations? m Let SR be a quantity SR alfa. (A [applied force(B)]-A [internal force(B)j) influences: Q+ m m — I + (iength(B), SR) Indeed, when we apply a force to an elastic object an imbalance between internal and applied forces will result in a length chenging. In particular, any quantity qualitatively proportional to the force imbalance has a positive "influence" on the length of the object. (The influence would be negative in a compressing process). Processes become active - during qualitative simulation - when the Individuals are instanced and the Conditions are satisfied. Processes communicate with other processes by means of Conditions (in input) and Relations/Influences (in output). The evolution of a physical system subject to a perturbation is represented by a graph in which nodes are active processes. The evolution is usually non-completely determined (from a node more than one transition is possible), due to the nature of qualitative reasoning. Quantitative analysis must be used when it is necessary to remove any ambiguity. This kind of representation makes explicit the causal relations between events, and can be used not only to reason about the effects of some mechanical action (qualitative simulation) but also to find an action that achieves some desired effect. Furthermore, the explicit relations in processes and in individual views provide possible constraints to be taken into account in the course of performing assembly tasks. For instance, if we want to keep stretched an elastic object, from the above description we can know that it is necessary to "apply a force such as to equilibrate the internal force". 4.4. Law-as-Process vs. Law-as-Constraint Processes embody the (qualitative) notion of Causality. Processes are the mechanisms that directly cause changes in objects. This view of physical phenomena is in contrast with the 9 Of course, this assumes a mono-dimensional description ol an object, thet is something thet can only ba pulled or pushed. In a general three-dimensional situation the domain of possible bahaviors becomes much richer (bending, twisting, lolding,...).

 traditional attitude of physicists who aim at capturing a generality of behaviors by means of "physical laws". Let us consider, for example, the Hooka's law: f « k (x-Xq) that states - in concise, elegant terms - the proportionality between the lengthening of an elastic object (beyond a given rest length) and the applied force. This law is a constraint between two variables - f and x - and it says nothing about causality: It allows, indifferently, to compute x from f or f from x. The semantics and the conditions of applicability of a law of this kind are either expressed informally or assumed implicitly to be known by the scientific community. In other words, the traditional formulation of scientific laws is a highly distilled formulation that takes for granted exactly what qualitative reasoning attempts to spell out. Qualitative physics and qualitative process theory only describe the form of qualitative dynamics. However, understanding human movements and purposive actions requires a representation of shape and space (spatial reasoning) which is far from being integrated with that kind of theories. 4.5. Different Paradigms of Qualitative Knowledge The type of knowledge that is captured by a Qualitative Process Model is very analytic, although it is qualitative - not quantitative in the traditional scientific terms. This is the knowledge, for example, of technicians who can "read" schematic engineering diagrams of electric or hydraulic systems for maintenance or trouble-shooting. This is certainly a paradigm of qualitative reasoning, but it is not the only one, because the knowledge required In many skills of everyday life may be very varied. In particular, we may consider the following paradigms which, in addition to the "Technician Paradigm" outlined above, single out many aspects of every day skilled motor behavior: 1. The Driver Paradigm. Different types of knowledge are required In navigation. One is the ability to "read" maps, that is a symbolic description with which it is possible to perform qualitative simulations similarly to the technician paradigm. But this Is only a small part of the skill. Any driver is indeed supposed to constrain his behavior according to the acknowledged Traffic Rules (the supervisory part of the skill), while focusing his Attention on the actual Traffic Conditions (the sensory - motor part of the skill). 2. The Player Paradigm. This is related, in general, to the problem of how performing efficiently a team work 10. This requires some knowledge which Is shared by all the players: the knowledge of the play rules (a set of axioms which cannot be modified) and the knowledge of the play schemes (a set of rules which are chosen by the coach, i.e. the planner). 10 Our nation ol team work is far away Irom what is known as Team Theory, a chapter ot operationa research which takes into account the cost ot communication in optimal decision making jMarschak and Radnor 1971).

 Moreover, the coach will assign a role to each player, taking into account his particular skill, and the player will have to use this individual knowledge, in addition to the general knowledge, while performing in the field. 3. The Cook/Bricoleur Paradigm. As a plumber must be able to read hydraulic diagrams, a driver must be able to read maps, and a player must be able to read tactical plans, so a cook must be able to read recipes and a bricoleur assembly instructions. However, this is only the starting point which can be successfully overcome only if the actor is not "clumsy", that is if he can perform precisely a number of actions (picking, pushing, pulling, cutting, pouring, etc.) in space and time, while taking adequately into account the shape of objects and their physical properties. The brief (qualitative) description of the previous paradigms well epitomizes the multiplicity of "dimensions" which span the "knowledge space" of common skills and makes it clear that it is impossible to reduce them to the same denominator. Specifically, we deem it useful to single out the following couples of opposites: 1. Synthetic vs. Analytic. The knowledge required in qualitative process theory, for example, is very analytic. Accordingly, this type of qualitative reasoning is very detailed, it is able to provide explanations of every facet of a process and it is necessary when reading a diagram. However, this is useless in most other cases where a small set of behavioral rules (in navigation, in team work, in assembly) may well embody very lengthy explanations. On the other hand, also in these cases, unexpected occurrencies may require a detailed qualitative analysis. Therefore, the ability to shift from synthetic knowledge to analytic knowledge, and vice versa, is quite necessary. 2. Analogic vs. Symbolic. In navigation and assembly work, certain kind of knowledge is naturally originated in analogic terms (distances, areas, volumes, curvatures, normals) which allow to perform the measurements required by many skills (reaching, inserting, pushing, etc.). Other knowledge, however, is naturally symbolic ("object A is above object B", "push object A until its left corner touches the yellow face of object B", "stop stirring the soup when it boils"). There is no rigid hierarchy between the two types of knowledge: they are complementary. 3. Prescriptive vs. Suggestive. Certain kind of knowledge Implies a rigid sequence of actions. For example, the assembly instructions of an assembly kit: They are prescriptive. Other kinds of knowledge, e.g. the knowledge of the traffic rules or the laws of robotics by Asimov, constrain behavior indirectly, in a supervisory way. Prescriptive specification of action is accurate but unflexible. Suggestive specification, on the other hand, embodies more "wisdom" about the global aspects of a performance, but is of little help in actually carrying it out. Purposive action is an open system with many actors of different types (analogic, symbolic, prescriptive etc.) - a system which may have different configurations " and which must be able

 to switch among them. Naturally, the self - configuration of an actor according to the needs of a particular situation cannot be unique. It is a choice - a subjective choice - which manifests the personality of a skilled actor. Purposive action cannot be objective and deterministic, it cannot help displaying that elusive mixture of genetic code, education, and experience which forms human personality. We terminate this section expanding one of the points outlined above: analogical vs. prepositional. 4.5.1. Prepositional vs. Analogical In most real actions, the interplay among the different components considered above is apparent. A good example of this situation is the soccer game, where the role of the planner is taken by the coach. His strategic and tactic decisions are crucial for a clever performance. However, the game is played, in reality, by the "players' and without their "skills" the subtlest planner would fail miserably. The soccer metaphor points out the interdependence between non deterministic planning and skiil. The knowledge or intelligence which is necessary to achieve the goal is shared by the planner and the players. The distinction between one planner and many (concurrent) players offers a general framework for trying to formalize a number of significant paradigms. The (one) planner, on one hand, emphasizes the fundamental need of a unitary focus of action. The concurrency of the multiple players, on the other hand, is the basic source of a smooth, continuous performance. The distinction above also corresponds to the distinction between a type of knowledge which is mainly "prepositional" and a type of knowledge which Is mainly "analogical". This distinction has been discussed in depth, among others, by Johnson - Laird [Johnson-Laird 1980] who has developed a theory of mental models: A mental model represents a state of affairs and accordingly Its structure is not arbitrary like that of a propositional representation, but plays a direct representational or analogical role. Its structure mirrors the relevant aspects of the corresponding state of affairs in the world. On the contrary A propositional representation is a description ... It Is true or falee with respect to a mental model of the world. The typical operation is geometric measurement, in the former case, whereas it is logic inference in the latter case. From our point of view (the computational analysis of action) the controversy between propositional vs analogical modeling, whloh has divided cognitive scientists [Anderson 1976, Pylyshyn 1984, Shepard and Cooper 1982] is not very important. Both are likely to play a role and we agree with Johnson - Laird that one is not reducible to the other except in " With actor configuration we mean a specification of a set of actors and ol the types of messages that they ex-chenga.

Cognitive Modeling of Purposive Actions 227 a trivial sense. We would also like to point out that the kind of mental/analog models that we are considering are not "mental images" but "mental solids" [Morasso, Sandini 1985). Mental Images are appropriate in iconic matching operations which may underly perception, whereas mental solids are necessary to estimate spatial relationships which are the source of skilled performance. Furthermore, we may observe that in this context "learning" can be formulated as a process in which a planner first figures out the skeleton of a skilled action, directly controlling the sequence of primitive actions and monitoring the results (the sequential control guarantees the coherence/unity of the performance but cannot provide the fluidity of concurrence) and only afterwards distributes the tasks to the players who will take care of the necessary message passing for achieving graceful performance. As a general framework, an action planning activity can be viewed as a process characterized by: 1. Large use of qualitative properties about both action and environment. 2. Use of skills as analogical models to make deductions not possible otherwise. 3. Generation of a composition of skills to solve the problem. 4. On line intervention to solve asynchronous problems, errors, etc. 5. PLANNING In the classic Al formulation, planning is rigidly placed at the top of the hierarchy, with world modeling and skill considered as low-level, Intelligence - free operations. In other words, the classic planning scenario is one in which the goal can be uniquely expressed as a "state" and it is perfectly known together with the initial state. Two examples of this situations are the following: 1. Path planning on a road map, from an initial town to a target town. (The planning problem is reduced here to exploring the road graph and to find the shortest route - the plan itself is the sequence of roads which must be covered.) 2. Assembly of parts from an assembly kit on the basis of an assembly model. The latter problem, in particular, lends itself to a STRIPS-like formulation [Fikes and Nilsson 1971, Nilsson 1980], where each action is expressed as a "rule", whose application is conditioned to a "precondition list' of predicates about the working environment and the actor himself (the "working state"): if the preconditions are met, then new statements are added to the working state (from an "add list" of rules) and other statements (stored in a "delete list" of the rule) are deleted from the current working state. A STRIPS-like planner needs an explicit specification of the Initial and final working states and then it looks for the set of rules that, applied in sequence, transform the Initial state Into the final state. A great deal of research has been focused on the alternative techniques for constructing the sequence of actions (e.g. forward or backward techniques), with the Implicit assumption that the characterization of the working state and of the

precondilion/delete/add lists is a minor problem. In realistic situations, this point of view is deficient. Let us consider some paradigmatic cases, which can help to outline a few general concepts. Missing information In many cases, the perfect knowledge assumed in classic planning is certainly false: 1. Path planning in a cluttered environment - which does not allow a complete view and is at least partially unknown - or in a generic environment (e.g. an "office") which is known only "by default". 2. Assembly of parts with an imprecise assembly model and/or with uncertainty about the presence of all the necessary parts. In both these cases it is crucial to use a kind of object modeling which is both multiple- definition and incremental. TimB varying environment The working environment can vary over time (in addition to the changes due to the programmed actions) as a consequence of either autonomous processes (e.g. a conveyor belt) or the actions of another actor (e.g. the two arms of a robots or two robots acting concurrently). The kind of knowledge which is particularly important in this case is the capability to integrate "events" in the planning process and to manage a significant "message passing" activity with the other actors. Actions, when performed, modify the state of the world, and that must be taken into account during planning. This is called the Frame Problem in Artificial Intelligence [Minsky 1975, Nilsson 1980], and adds a new requirement: the need of representing the effects of actions. Competition/Cooperation A competition is a particular purposive action where the goal is to win (or to get a draw) against a competitor. In most competitions (or games) the classic planning formulation is meaningless because it is impossible to find a plan that guarantees victory whatever the adversary does. The only alternative is to figure out a strategy and a tactics, i.e. a set of rules which, though not guaranteeing the victory, are likely to end up with a victory in most of the cases. The strategic rules are the framework which remains constant during the game, whereas the tactical rules will be adapted during the play. Furthermore, this kind of concept is not restricted to competition, but it remains valid in other kinds of co-operative actions (e.g. in a symphonic performance or in complex skills such as piano playing or tennis playing).

Cognitive Modeling of Purposive Actions 229 Planning Paradigms Among the drawbacks of the family of STRIP-like planners, speed - the unnecessary performance of lengthy searches - is one of the major ones. Different techniques have been considered to face the problem, for example learning methods based on the concept of analogy [Fu 1971, Tangwongsan and Fu 19791 or on schemes of "procedural learning" [Siklossy and Dreussi 1973] which enable a robot to accomplish tasks sequentially according to their partial ordering. The first kind of learning, in particular, corresponds to classroom learning - learning with supervision - where a teacher provides students with the procedural knowledge of some examples: The learned information is considered as an experience that will, in turn, be used to speed up the planning process. Apart from the learning concept, more efficient planning techniques than those formulated in the framework of theorem proving have been investigated, focusing on powerful heuristic control structures which go beyond the basic recursive control structure embedded in languages like LISP - control structures that may provide mechanisms for recovery from bad choices, for creating and invoking demon processes, for maintaining multiple world models, establishing dynamic hierarchies of goals and subgoals, and so on. A family of packages has been produced, such as PLANNER [Hewitt 19721 and CONNIVER [Sussman and Mc Dermott 1972], and in particular a planning system for robot construction tasks - BUILD - has been developed [Fahlman 1974] which, using a CONNIVER - based powerful control structure, can incorporate a number of sophisticated construction techniques in its plans: preservation of pre-existing structure in the final design, pre-assembly of movable structures on the table, the use of extra blocks as temporary supports, the use of counter-weights in the course of the construction and so on. The emphasis on the hierarchical aspects of planning has been put by Cohen and Feigen- baum [1982] and by Sacerdoti [1977], The NOHA planning system, in particular, is a system in which plans are partially ordered sequences of actions, represented as graphs: the graph nodes are either actions or special operators - split and join. Changing level of abstraction in NOAH means expanding a node in a subgraph. For instance, the expansion of the node corresponding to the action "achieve (festened A B)" determines a subgraph, as follows: level 1: achieve (fastened A B) level 2: (get A) (get screw) split join split join (fasten A B) (get B) (get screwdriver) Representing plans as partially ordered sequences of actions, as in NOAH, makes it possible to include concurrent actions. Synchronization problems can be stated in terms of constraints. In an assembly task a typical concurrent operation is one in which two hands are needed. (Usually, one hand holds a piece and the other performs a co-operating task.)

 Planning with constraints has been explored by Stefik [1961] in the MOLQEN system. Although MOLQEN was developed for planning gene - cloning experiments In molecular genetics, those planning concepts are relevant for eutomatic assembly because, as we have seen before, constraints play an important role within the context we are examining. In MOLQEN constraints are Interpreted in terms of: 1. Elimination Rules: conditions to be satisfied for object or action selection. 2. Commitments: a constraint is a partial description of an object or an action (least- commitment approach, i.e., deferring decisions as long as possible). 3. Communication Medium, for expressing interaction between subproblems. Planning with constraints in MOLQEN means to perform the following operations: 1. Constraints Formulation, i.e., adding new constraints; 2. Constraint Propagation, i.e., inferring new constraints from old ones in different parts of the plan; 3. Constraint Satisfaction, i.e., finding values for plan variables so that constraints are satisfied. Planning and Concurrency The Hewitt's Actor Systems suggest a general framework in which a Plan is modeled as a set of actions to be performed according to a (potentially total) concurrency Instead of a sequence. We might even hypothesize a planning strategy which proceeds Incrementally, by firing initially a number of purely concurrent actions and by constraining them subsequently, when critical situations are recognized which require a sequential synchronization. This Idea Is homogeneous with both the metaphor of "action as a team" and the principle of "self-similarity": A plan is, at any level of detail, a team of Actors, each one paying attention to specific aspects of the goal, according to his competence. (Consider, for example, "pick up", "walk", "avoid", "measure", "stay away from" and so on.) The Idea of having simultaneously active different parts of a whole action might fit many "short term" models of purposive actions. As an example, let us consider a task like Pick up a part and put It inside a box, holding It horizontal. If the part falls, then pick it up and go on. Of course, avoid collisions, please. In this short - horizon activity, the following concurrent actors may solve the problem: (1) A "reach/pick up" actor. (2) A "put down* actor, waiting for a signal from the former actor or otherwise checking for a constraint to be verified (the part In a certain position). (3) A "keep horizontal" actor, concurrent with the first one, terminating his activity after a message from the others.

 (4) A "collision avoider" set of actors, watching different parts of the body. (5) A "recovery actor", quietly ready for a possible "fall" - an event that would drive the arm back to a position where the other actors could resume their control. This script might be the actor-like formalization of a fragment of a whole plan, conceived by means of symbolic/propositional reasoning. Again, we emphasize the double nature of planning (analogic vs. propositional): We deem, indeed, that generic plans require both components and for any specific plan the most convenient "cocktail" Is a matter of experience and taste - a still open and stimulating research field. Let us now consider in more detail the planning issues with the

aid of an example: the assembly paradigm. 5.1. The Assembly Paradigm As any problem solving task, devising and performing an automatic assembly requires a suitable representation of the problem, which involves (i) a representation of the state of the environment, (ii) a knowledge base of the rules governing the environment and of the possible actions, and (iii) a strategy to solve the problem. Furthermore, both the initial and the final states of the environment are to be known, the latter being the goal of the assembly problem. 5.1.1. Problem Representation Starting with the environment, a typical assembly task has to deal with a set of pieces to be assembled, a set of tools, a working plane, a manipulator and a partially assembled part (which may not be present at the beginning of the task). The state of the environment neede to be conveniently represented in order to solve the problem. The state is determined, in its turn, by a set of parameters, related to each component of the environment. Parameters can be: 1. The position and the orlentetion of a single piece. 2. The geometric and structural configuration of a composite piece, be it an assembled part, a tool, or the manipulator itself. 3. Kinematic quantities (velocities and accelerations) related to the manipulator's motion. 4. Dynamic quantities (torques and forces) applied to the manipulator. 5. The size and shape of objects and, complementarity, the size and shape of the free space within the working environment. Let us consider, in particular, the first two points. Position and orientation of rigid pieces are described by six parameters in 3-D space. Composite pieces can be described geometrically in terms of mutual position and orientation of component pieces and, structurally, in terms of spatial symbolic relationships, such as against, coplanar, fits etc., that express how features are related to a body. The geometric modeling aspect of this topic is further analyzed in Part III.

 The interplay between geometric modeling and symbolic geometric reasoning was investigated by Popplestone et al. [1978], who proposed an algebra for describing configurations of assemblies. They use the operator trans(x,y,z) to express translation with respect to a given reference frame and the operators twix(theta)/twiy(theta)/twiz(theta) to express a rotation by theta along the x, y, and z axes, respectively. They give also a set of rewriting rules that may establish the equivalence between different expressions. Relationships among features of a body are expressed symbolically by means of the symbolic spatial operators previously mentioned, a feature being a planar face, a cylindrical shaft, a cylindrical hole, an edge, a spherical face, a vertex, etc.. These relationships can be mapped into algebraic expressions. For instance, if B1 is an object with a feature F1 - a shaft - and B2 is another object with a feature F2 - a hole -, then the statement "F1 fits F2" can be expressed algebraically as follows: p2-(f2)~1 twix(theta) trans(x,0,0) fl p1 where: p1 is the position/orientation of B1 with respect to the environment, 11 is the position/orientation of F1 with respect to B1, trans(x,0,0) is a translation to the tip of the shaft, twix(theta) is a rotation around the shaft axis, f2 is the position/orientation of F2 with respect to B2, p2 is the position/orientation of B2 with respect to the environment. The rules governing the environment are the structural and physical laws that objects ere subjected to. Such laws pose a set of constraints on the possible actions that can be performed. Structural laws usually limit the access to parts of the objects or of the space. For instance, you cannot pick up a part if you have other parts that conceal it or limit its free motion and you cannot place a part in a region of the workplane which is already occupied by other objects. Furthermore, structural laws determine the possible ways in which parts can be matched during the assembly. For instance, if two parts are to be connected with a set of screws, their faces must contact and holes must correspond. Structural rules must be taken into account in planning actions as a set of constraints to be satisfied. Physical rules impose further constraints to an assembly task. For example, gravity strongly limits the possible ways in which pieces can be handled. We have to take care of the fact that parts need to be supported before being connscted to other parts, and often we cannot use a partial assembly or the workplane as a support, as it happens, for instance, when the parts are to be mounted in a vertical structure. That usually requires another hand to hold the part12. In order to handle the parts properly, we need to know motion and dynamic laws. For instance, some pieces can be moved around a workplane by simply pushing them: that means, getting In contact with them and providing a force directed toward both the object and the desired direction, and not otherwise. The kind of motion, however, depends on the surface and on the type of contact. Some parts can store energy, and therefore need particular caution. 12 In Industrial automated environments, planar supports are often present designed to temporarily hold parts during assembly: They act as "passive" hands lor thet special purpose, that Is they are an example ot the kind ot ad hoc solutions currently adopted in industrial robotics. A second hand would ba, in this case, a more general solution.

Cognitive Modeling of Purposive Actions 233 Springs and elastic pieces, for instance, can chenge their shape upon pressure, responding with opposite forces. In a previous section we discussed the methods which have been developed for a quelitative representation of this kind of phenomena. The initial state of the environment is one of the possible states and can be described by the values of a set of parameters, as it has been remarked above. The final state - the goal - can be specified in a similar way, possibly with a partial description, containing only some relations among parameters. For example, in the language RAPT [Popplestone et al. 1976] the assembly task is specified in terms of a set of spatial relationships between some features of the objects that must hold in the final state. Other relationships are inferred analytically and a possibility tree is constructed in the case of an incomplete specification. 5.1.2. Problem Solving Robot planning is a special kind of problem solving. A simple assembly task, for example, might be planned by means of a sequence of single assembly operations of the following type: moving, grasping, releasing. what is known as a "pick_and_place" paradigm, typically fit to the STRIPS formulation. STRIPS-like planners use a symbolic representation of the state of the world in terms of formulas of predicate calculus. A list of preconditions (formulas that must be true in order to perform the action) is associated with actions, together with an add list (formulas that are true as a consequence of the action) and a delete list (formulas that are no longer true after the action is complete). For instance, for a simple assembly task in the "block world", the following actions can be defined [Nilsson, 1980): pickup(x) P & D: ontable(x), clear(x), handempty A: holding(x) putdown(x) P & D: holding(x) A: ontable(x), clear(x), handempty stack(x.y) P & D: ho)ding(x), clear(x) A: handempty, on(x,y), clear(x) unstack(x,y) P & D: handempty, dear(x), on(x,y) A: holding(x), clear(y) where P, D and A stand respectively for "preconditions, "delete list" and "add list". The goal

234 G. Adorn i et a I. statement is represented by means of logical calculus formulas. For example, the task of building a bridge - shaped object in which two cylindrical columns (C1 and C2) support a beam (B) may be stated as (uprlght(CI) & upright(C2) & on(Cl,plane) & on(C2, plane) & on(B,Cl) & on(B,C2)& equal(distance(C1 ,C2),length(B))). STRIPS employs a resolution theorem prover and a means - end analysis strategy to guide It to the desired goal satisfying model 13. Other search strategies have been investigated, for example a goal regression mechanism for circumventing goal interaction

problems [Nilsson 1980]. 6. THE NAVIGATION PARADIGM We terminate this chapter with a discussion of navigation as an action planning process in whloh partial information, model updating and run-time mechanisms need to be integrated. Navigation of an actor (or more actors) in a cluttered environment is akin to skilled manipulation and presents most of the aspects previously emphasized. It requires (i) to model the working space in order to avoid collisions with obstacles, (ii) to plan a probably safe path from a starting position to a target position, (ill) to "tune" motor skills of the actor. These global aspects must also be Integrated with the capability to deal with the differential, local, unexpected/ unaccounted aspects of the action to be performed. In the literature we may find two main classes of algorithms: (i) free space representation in a polyhedric world [Lozano Perez 1962, Brooks 1983, Canny 1984], (II) penalty function [Udupa 1977, Khatib 1980, Hogan 1982). In both cases, however, the complexity of the navigation task increases exponentially with the articulation of the world model because planning is performed exploiting the full three-dimensional representation of the world. The approach which is proposed here - the SMASH project (Soft Maneuvering among SHapes) - is aimed at distributing the navigation complexity into different elements: 1. A general 3-D Geometric Modeler, which needs not be restricted to polyhedric objects. 2. A Navigation Expert which, given the general world model, the actor model and the task to be performed, generates a simplified representation of the environment. This representation is essentially 2-D augmented with 3-D attributes; we call such a representation Augmented Skeletal Representation (ASR). 3. A Planner which, given ASR and a description of the actor's skills, provides a strategy to accomplish the task. 13 The means-end analysis i« the process ol identifying the "dKtertnce" between the current stele end the goel stats and ot choosing the action (or rule) that reduces such difference by the greatest amount.

 4. An impact avoidance mechanism which operates on local information acquired during the actual navigation - it implies a perceptual process monitoring the actor/environment relation. Figure 1 shows an outline of the system's architecture. Synthetically, the philosophy which underlies the SMASH project is a divide and conquer approach, where each component of the system can be accepted to fail in several occasion, without causing the overall performance to collapse. 6.1. Modeling the Environment The concept of environment we are using is -

roughly speaking - a set of objects plunged in a limiting volume. Objects are obstacles and space among them is free space. Given a model for limiting space and obstacles, free space can be computed as: free space « limiting space - (union of obstacles) This computation can be performed by means of a modeling scheme based on Boolean operations among shapes, like, for example, Constructive Solid Geometry [Requicha 1980]. In fact, we are actually using PADL-2 [Brown 1982] as a modeling tool, but it is worth pointing out that the proposed approach is not dependent on this choice; CSG makes just easier computing intersections, which are required, as explained later, to model paths. Furthermore, SMASH can deal with general object models, whereas PADL-2 has a limited domain (i.e. it relies on a limited set of primitive shapes). Let's now consider the problem of how to obtain information about the topology of free space. Significant topological information can be captured by sectioning the environment according to a significant plane (slicing): for example a plane parallel to the floor (figure 2). Slicing should also be done at a meaningful height with respect to the moving actor. Considering a human meaningful heights may be, for example, 10 cm, 1 m, 1.70 m (foot-level, trunk-level, NAVIGATION EIFERT <y ERROR DETECTION O 0 <? IMPACT AVOIDANCE <? Figure 1. Outline of SMASH'S architecture world

 and eye-level, respectively). Let's now assume to observe the sliced environment downward in a direction normal to the floor; what we see is a binary Image containing sections of obstacles and free space. (From a practical point of view, these images can be obtained in PADL-2 by computing the intersection between the PADL world model and a thin parallelepiped and by storing in a frame grabber the corresponding image.) The structure of free space at a certain slicing height is captured by the skeletal representation introduced in the first Chapter of Part III. However, the representation of a single slice is likely to provide too little information, because it ignores the third dimension: we need to know, at least approximately, what's going on over and under the slicing height for the whole extent of the actor. This can be achieved in two ways: (i) by means of multiple slices (multiple SR's) whose topologies should be correlated in some way or (ii) by enriching with 3D attributes the SR of one slice (Augmented Skeletal Representation: ASR). The second approach was followed since it appeared to be more promising. Given the SR at a certain slicing height, the corresponding ASR can be obtained by considering, for each sample of the skeleton, a line normal to the slicing plane and by computing intersections of this line with obstacles: This computation can be done again with PADL - 2, using a thin cylinder or parallelepiped as a * probe*. The ASR is then defined as SR plus a positive height function and a negative height function, where both functions, in analogy with the thickness function, are scalar functions of a variable running on the medial axis - one function measures the distance of the medial axis from obstacles above and the other from obstacles below the slicing plane. In this way the ASR can provide information about holes, arcs, galleries, protuberances of obstacles and so on. It Is worth remarking that ASR can give just a rough idea about topology of paths. In fact, we wish to avoid the unnecessary complication of modeling paths too precisely. Our goal is to provide a simple representation of the main structure of the environment to be associated with a Figure 2. Extraction of a section of a world model, by means ol a "cutting plane*, parallel to a "door plane*. The section determines a binary Image, from which closed contours are first extracted and then smoothed.

Cognitive Modeling of Purposive Actions 237 set of low-level procedures, able to make corrections on-line. Several types of errors may, in fact, occur, such as: (i) errors due to neglecting the fine structure of the boundary of obstacles, (ii) errors due to inaccuracies of the ASR, (iii) errors due to deficiencies of the 3D geometric model. All kinds of errors can be corrected by means of local manoevering of the actor and/or by replanning as discussed later. 6.2. Modeling the Actor Since actors may be viewed as objects which move among obstacles, then it is reasonable to model actors not only on the basis of geometric features but also of their motion capabilities. From a geometric point of view, modeling can be done, like for all shapes, at different levels of details. At the lowest level, an actor can be represented by means of its three-dimensional extent, i.e. the enclosing parallelepiped of minimum volume; such a model, though rough and imprecise, is appropriate for sketching navigation plans on the basis of simple heuristics. Other more detailed models could be used for finer specifications of plans in critical sections. A way of modeling motion capabilities is to specify how movements affect the geometry of actors when they are performed. A human, for example, can stand, walk, crawl, stretch arms sideways, sit, lay on the floor, coil up and so on - each of these postures being characterized by a different 3D extent and shape. In SMASH, the model of an actor is simply its extent and a set of modifying rules mimicking motion capabilities. Such a model allows to adapt the actor according to the kind of obstacles encountered: it can then get sideways into narrow passages, pass under hanging objects and so on. The model of a human-like actor with a few motion skills can be presented as follows, by means of a PROLOG formalism: standard([180,50,30]). /* dimensions of the 3-D extent (parallelepiped) '/ actual([180,50,30]). can_slt(man). can_stretch_arms(man). can_lay(man). can_crawl(man). can_go_on_all_fours(man). change(X): - can_sit(X), standard(Y), mul(Y,[0.72,1,2.33j,Z),asserta(actual(Z)). change(X): - can_lay(X), standard(Y), mul(Y,[0.16,3.66,1.66j,Z), asserta(actual(Z)). change(X): - can_stretch_arms(X), standard(Y), mul(Y,[1.25,0.8,1 j,Z), asserta(actual(Z)). change(X): - can_crawl(X), standard(Y), mul(Y, [0.38,0.9,4.j,Z), asserta(actual(Z)). /* mul(X,Y,Z) is a rule which multiplies each element of X by each element of Y and appends the result to Z */ TABLE 1

238 G. Adorni et at. 6.3. The Navigation Expert The navigation expert (NE) computes, initially, the ASR graph from a 3 - D world model, taking Into account the task and the actor model. At this stage of our research the task is simply "to reach safely a target position" and it does not affect significantly the kind and the level of detaii of the information stored in the ASR by the NE. In fact, richer tasks can be formulated, like, for example, "to move an object from one place to another place" in which the size of the object to be carried by the actor can affect the navigation strategy and the obstacles' layout can change after the task has been accomplished. This kind of problems and the knowledge necessary to handle them will be explored in the near future. To build the ASR, NE performs the following steps: 1. Choose the slicing height on the basis of the physical model of the actor. 2. Compute the SR for this section. 3. Plan a tentative path from the SR. 4. Compute the ASR of the selected path. This means computing Intersections with obstacles over and under the slicing height and storing in the ASR graph all critical points, i.e. those locations for which the height of free space is less than the height of the actor. 5. Check if these critical situation can be overcome by means of the motor skills of the actor. If this is the case, the resolving motion is stored for each critical point, otherwise a no-way mark is put in that path and a new path - planning attempt is made. The ASR is composed of a SR plus critical 3D features and manoevering attributes. This representation Is actor-dependent and task-dependent, that is different ASR's can be obtained for the same environment but with different

actors or tasks. Moreover the representational burden related to the third dimension Is limited only to the potential trouble spots. Once a relatively safe path has been selected, interferences with the real world are corrected by impact avoidance mechanism and are protected by the compliance of the actor. When situations of deadlock occur and are detected by the Impact avoidance mechanism (the actor cannot proceed any further), NE uses this new evidence to update Its knowledge, by adding new constraining attributes to the ASR. The planner is then re-invoked by NE and planning can be started again on a better knowledge base. 6.4. Planning The Path: The 'Which and How' Approach As already stated in the previous section, the path-planner is invoked by NE: It provides, Initially, a promising path on the basis of 2-D information according to some predefined strategies like, for example, "select the shortest path", "select the safer path" and so on. It is re- invoked every time critical difficulties are encountered either during analysis of the third dimension, or at run-time.

Cognitive Modeling of Purposive Actions 239 Finding the path is not, by the way, the only planner's duty; In fact, once the path has been definitely selected, the actor must choose an efficient way of traveling on it - the trivial solution consisting of following the medial axis. Another criterion, which minimizes the covered distance, is to keep on the inner sides of the curves. This heuristics is captured, In part, by the following procedure: begin smart_drive(initial_point,final_point,trajectory) if IN_FREESPACE(initial_point,final_point) then trajectory « straight line; return(trajectory); else find a sequence of target points {T1 Tj,...Tn) such that: T1«inltial_point, Tn «final point, IN_FREi5PACECTi,Tj+1) for any j trajectory « move towards Ti until IN_FREESPACE(current_positJon,Ti +1) return(trajectory) end smart_drive where IN_FREESPACE(A,B) is a predicate which is true if the straight line connecting A and B does not interfere with the obstacles. The output of this heuristics is a polygonal which can be arbitrarily refined and smoothed recursively increasing the number of target points. Figures 3 and 4 present some simulation results. Figure 3 shows the skeleton of free spaoe for a section of a 3 - 0 environment; figure 4a shows the polygonal found by the above algorithm for a cyilndric actor with radius « 20 pixels; figure 4b presents the polygonal between the same initial and final point, but for an actor with radius « 10 pixels. It is worth noting that the true trajectory of the actor will result from applying to these poiygonals the impact avoidance mechanism described in next section. 6.5. Compliance and Impact Avoidance Compliance and Impact Avoidance (IA) are two complementary safety mechanisms that guarantee soft interactions of an actor with his environment: Compliance provides soft impacts and IA prevents impacts in a soft, compliant way. Compliance oharacterizee the way In which a manipulator reacts to deviations from intended trajectories (it can have both elastic and viscous components). IA can be represented by a repulsive function which depends on the distance of the actor from an approaching obstacle and on the speed of the approach. To begin with, let us consider the following point model of an actor: p(t) planned trajectory p'(t) planned velocity d(t) distance to the nearest obstacle n(t) normal unit - vector to the obstacle

 We can suppose that d and n are perceptual variables, which result from an examination of the surrounding space of the actor on an expanding sphere. The reader may note the similarity of this Idea with the idea underlying the medial axis transform In the analysis of shape. IA can be implemented by correcting the planned trajectory with a vector which points away from the obstacle. In particular, we used the following repulsive velocity: v - -BtdXP' ■ n) n 14 which Implements a distance dependent viscosity. The repulsive gain g can be chosen In such a way that the apparent viscosity becomes very great when the actor is close to the obstacle and dies out after a certain safety distance. In particular, we can use the following function: g-BW-e-***18 where delta (which has the meaning of a space constant) can be chosen by trading off the danger of an impact with the burden of the perceptual analysis. Such a correction strategy, in ideal conditions, guarantees a perfect protection from impacts. Let us now consider the model of an actor as an unconstrained rigid body, I.e. as a rigid body with six degrese of freedom in the case of spatial movements. We may suppose that an intrinsic system of reference is attached to the body and that the planned movement Is a translation of the origin and a rotation around it: p' planned translation velocity vector omega planned rotation velocity vector Figure 3. Skeletal Representation ol free space for a section ol the working environment. <> denotes time derivatives, <x> vector products, and <> scalar products.

 Let us also denote by R the rotation matrix of the local system of reference with respect to the environment. Again, we assume that perceptual processes can provide some estimate (within a given spatial horizon delta) of the distance and of the normal unit-vector (d, n) of the closest obstacle, together with a specification of the vector b which joins the origin of the local system of reference to the point of the boundary at minimum distance. The repulsive velocity, in this case, must be calculated by taking into account both translation and rotation: v « -g(d) ((p' +omega x R b).n)n 1S In order to implement the correcting action, it is necessary to decompose the repulsive velocity into two components (v1 and Vg) to be associated with the translational and rotational corrections, respectively. In particular, It seems natural to choose the two following directions: (i) the direction given by b for v^ and (il) a direction normal to b and coplanar with (b, n) for v2- v1 gives directly the necessary correction for p'. The correction vector for omega can be computed from v_ as follows: correction(omega) - (R b) x Vj/fb.b) (b, v2, and the correction angular velocity are an orthogonal triple of vectors and the modulus of the angular velocity is equal to the velocity divided by the distance). If the motion of the rigid body must be constrained, then in decomposing v we must take into account the nature of the constraint and, in particular the fact that whether it is hoionomic or not. However, instead of modeling this kind of problems, we rather like to focus on another extension of the IA strategy, namely the case in which the actor can be modeled as an articulated body, I.e. as a tree of kinematic chains, with links and joints, which can move In an anthropomorphic way. Figure 4. Planned polygonal trajectory from an initial location (left-up) to a final target location (right- down). We assume a oyllndric actor with radius - 20 pixels in the left figure (a) and with radius « 10 pixels in the right figure (b). The expression in the inner couple of parentheses corresponds to the velocity distribution law in a rigid body.

 In fact, each link can be modeled as a rigid body and the articulated structure implies a distributed system of constraints among adjacent links. During a planned movement of an actor, each of his links can be guarded by an IA process similar to the one described for the rigid body model. It is important to realize that the computation of the planned motion of a link (translation and rotation velocities, in particular) Involves direct kinematics, whereas the correction due to IA requires inverse kinematics [Benati et al, 1983]. In other words, the required modification of p' and omega must propagate backward in the kinematic chain, involving as many joints as it is required. In general, this implies a six-degree-of-freedom problem, because an arbitrary roto - translation is characterized by six independent variables. As a consequence, the kind of required inverse kinematic computation depends on the location of the impending Impact: if it affects the hand (or the foot), then the classic inverse kinematics of the arm (or of the leg) is what is needed, whereas, if it affects a link closer to the trunk (i.e. the father node of the kinematic tree structure) then it is necessary to invoke also a motion of the trunk which, in turn, may trigger a walking pattern. It should be evident, at this point of the presentation, that an IA strategy for a human - like articulated mechanism must be integrated with a formalism for expressing complex geometric structures and motor coordination patterns. Another kind of essential Integration, i.e. the integration between perceptual and motor processes, has already been mentioned, but it may be worth to enrich it with another useful anthropomorphic concept: attention. In our context, selective attention means that not all possible IA processes need to be active simultaneously at any one time, but only those which are expected to be critical. Activation of the relevant IA processes must be the result of planning, i.e. of pre-motor cognitive processes. Another linkage between motor and cognitive processes occurs in the opposite direction, when an IA process detects a "dead-end" situation, i.e. a situation which requires re-planning. This is the case, for example in the point model, of a planned trajectory which happens to be normal to an unexpected obstacle surface: in this situation, the correction mechanism would take the point to a stall, inhibiting it from reaching the target location. Cases such as this one can be detected by comparing the planned and corrective velocity vectors and by invoking the planner, when the two vectors tend to annihilate, and passing It the newly acquired information about the local geometry of the unexpected obstacle. A final comment concerns again the relation between compliance and impact avoidance: such a relation is characterized by a strong functional complementarity in the actor/environment interaction, i.e. in the phenomena that precede and follow impacts. If an actor can rely on them, then planning becomes much less critical, and the whole SMASH becomes a much more robust action.

PART III SHAPE Movement is the final result of a plan of interaction of the body with the environment. The initial prerequisite for the Central Neural Controller is the representation of space, of the objects, of their mutual relationships as well as of their relationships with the self. An important, although not the only, part of such representations is geometric in nature. For example, when pouring coffee from a coffee pot into a mug, the appropriate planning and execution requires to take into account the shape, position, and orientation of the coffee pot and of the mug: the shape, position, and orientation of the movements must match the shape, position, and orientation of the environment. Therefore, the geometric modelling of the environment and of the objects is a fundamental prerequisite of a global motor control system. In this part we explore a few aspects of this general problem in order to provide some useful methodological tools. In particular, the flrst chapter discusses methods for the analysis and synthesis of smooth planar shapes and the second chapter reviews some methods for the representation of surfaces and volumes. It is clear that the geometric representation of the environment is at the same time the initial step for the motor control system and the final step for the vision sytem. The iogic consequence is that such a representation must be "neutral", i.e. independent of the specific, idiosynchratic characteristics of both the motor and the visual systems. There is also ground to suppose that such a representation communicates bldirectionally with the motor and the visual systems. This means that the geometric representation of the environment does not drive the motor system in a purely top-down fashion but it is refined and updated by it during the motor performance. In a similar way, the visual system does not generate the geometric representation in a purely bottom-up fashion but it is driven by it, in a recursive way, taking into account at each instant the current representation.

This Page Intentionally Left Blank

PART III Chapter 1 ANALYSIS AND SYNTHESIS OF SMOOTH SHAPES

Salvatore Qaglio, Aldo Grattaroia, Una Maaeone, Pietro Moraaao

CONTENTS

Least Squares Estimation of Branch Descriptors Appendix C: SCULPTOR - 2 Commands

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliesco (Editors) © Elsevier Science Publishers B.V. (North-Holland), 1986 247 ANALYSIS AND SYNTHESIS OF SMOOTH SHAPES Salvatore Gaglio, Aldo Grattarola, Una Massone, Pietro Morasso Smooth shapes are defined as connected regularized point sets whose boundary is "smooth". We distinguish, in the structure of an object, between a "main structure" and the overlying "details" or "textures" and we propose a complementarity principle for the description of shape: "Shape demands description at different levels of detail: global descriptions require low detail whereas local descriptions require higher detail. The relative emphasis depends on the application". We suggest that the two entitles can be separated in terms of frequency analysis, the low frequency component being associated with the main structure, and the high frequency component with details/textures. The representation of shape that Is proposed consists of two complementary components: (i) the "skeletal representation" of the main structure, which uses the concept of Medial Axis Transform ,and (II) the "zero- crossing representation" of the details/textures which extends to the contours of shapes the theorem by Logan already applied to problems of vision. The two representations are discussed with regard to problems of analysis and synthesis. 1. INTRODUCTION A puzzling aspect of quite a few natural phenomena is that they display, at the same time, "smoothness" and "structured discreteness". This is the case, for example, of natural speech and handwriting, where the superficial continuous flow of Information is "obviously* underlled by a strongly structured mechanism (consider the notion of phoneme or grapheme [Flanagan 1982, Thomassen and Teullngs 1979, Morasso and Mussa Ivaldi 1982]). A model which fails to capture the essence of the relationship between smooth surface and structured interior is bound to perform poorly either In terms of naturaltty or of efficiency. We intend to explore such a topic, which is central to Artificial Intelligence, with regard to the shape of objects, particularly in the case of planar shapes. (However we think that the main concepts and methods can be extended to three dimensions.)

 A considerable amount of attention has been devoted to the computational treatment of shape in the field of geometric modeling. Among others, Requicha and coworkers [Requtcha 1980, Tilove 19801 vary carefully pointed out the importance of a well defined mathematical formulation of the ooncept of object for deriving meaningful computational schemes. They embedded the notion of rigid object In the topology of the Euclidean space - 2D or 3D - by defining regularized-sets (shortly r-sets). R-sets are characterized by the fact that they coincide with the closure of their interior, l.e they do not have isolated or dangling parts. Moreover, Requicha and coworkers developed a representation scheme called CSQ - Constructive Solid Geometry - which is based on "regularized" set operations (union, intersection, subtraction). Even if CSG is very attractive for its mathematical clarity, it fails to capture the notion of 'smoothness"; CSQ shapes are bound indeed to display sharp edges or comers, I.e. dlscontlnultlee. However, still In the framework of the Requicha formulation, we may define "smooth shapes" as connected r- sets (CSQ may generate disjoint r-sets) whose boundary is continuous up to a certain level: If the boundary Is expressed In terms of parametric equations, we assume at least continuity of the second derivative for pianar shapes (It determines curvature) and of the third derivative In the 3D case (it determines torsion and the Gaussian curvature). Our goal is to deal with "smooth r- sets", looking for their structure. An additionel intuitive notion that we deem important in order to understand shape is the distinction, in the structure of an object, between a "main structure* and the overlying "detaiis" or "textures". For example, different door keys might be characterized by a similar structure - an oblongated protuberance linked to a rounded body - but by different details - the teeth of the key. The profile of a mountain ridge will show, In different times of the year, different "textures" superimposed on the same underlying structure or the other way around If we consider different mountain ridges with a similar "foliage". We may introduce a complementarity principle for the description of a shape: Shape demands description at different levels of detail: global descriptions require low detail whereas local descriptions require higher detail. The relative emphasis depends on the application. The complementarity principle implies that the shape of natural objects cannot be described uniquely: it is possible indeed to go from the molecular to the macroscopic structure passing through a large number of significant levels of description. The suggestion that we put forward is that the two entities can be separated In terms of frequency analysis, the low frequency component being associated with the main structure, and the high frequency oomponent with details/textures. The representation of shape that is proposed consists of two complementary components: (i) the "skeletal representation" of the main structure, which uses the concept of Medial Axis Transform [Blum and Nagel 1976], and (ii) the "zero-crossing representation* of the details/textures which extends to the contours of shapes the theorem by Logan [Logan 1977] already applied to problems of vision [Merr et al. 1978]. The two representations are discussed in details in the following sections together with computational aspects for both analysis and synthesis of shapes.

Analysis and Synthesis of Smooth Shapes 249 2. GEOMETRIC CHARACTERIZATION OF THE SHAPE OF OBJECTS 2.1. Regional/Volumetric Characterization <RS> RS (r-set) is a connected point set in E2 or E3 which coincides with the closure of its Interior; RS's can have holes. For RS's to be appropriate models of rigid objects they must be congruent classes, i.e. their representation must be independent of translation and rotation. <SRS> It Is a 'smooth* RS, I.e. an RS whose boundary Is sufficiently continuous (In the case of holes, the boundary is not simply connected and the continuity condition must characterize separately each connected component). <DSRS> It is a "discrete" SRS: it may be obtained when a SRS Is sampled (e.g. by means of a rectangular grid) and it Is stored in a digital binary image. It is worth noting that If the sampling grid size is not small enough, the binary image might have "dangling' edges, a kind of pathology for which the notion of "regularized" set has been defined. The repolarization of a discrete set can be obtained by applying computationally the same definition of r - set, i.e. by means of the following algorithm: 1. extract the contour of the image 2. delete it "/it corresponds to consider the interior of the shape/* 3. extract the contour of the complement image 4. add it to the shape */it corresponds to compute the closure/* It is also worth noting that the binary image representation of a DSRS is a simple but very time efficient way to perform set membership classification, i.e. to test whether a pixel is inside, outside or on the boundary of a shape. 2.2. Continuous Boundary <B(SRS)> It is the boundary of a SRS, which is a continuous closed curve without self-crossing points. (If the SRS has holes, the boundary is a collection of disjoint curves of this type characterized by the fact that one of them contains all the others.)

250 S. Gaglio et al. Usually, the boundary Is expressed in parametric form as the hodograph of a vector function of an independent variable t, normalized for example in the Interval (0,1): (2.1) <B(SRS)>::- (r-r(t)-x(t)i + y<t)D The two components of r, which are both functions of t, are continuous and periodic. The boundary r is represented parametrically if there is a one-to-one correspondence between a certain value of t in a period and a point in the curve, the correspondence being continuous at each point of the period. This oonstralnt ensures that the curve does not have points of self - intersection and, consequently, it encloses a simply oonneoted plana region. An interesting oase is that in which the parameter of the vector function Is the arc length I of the curve. In such a case, the equation (aa) r - r(i) describes the motion of a particle along the curve with constant speed. For a closed contour the arc length can be computed from an arbitrary starting point In order to obtain the arc - length dependent parametric function of a curve r - 1(1) from a given parametric function r - r(t) of the same curve, with an arbitrary parameter t, it is possible to use the following transformation: t (2.3) I - INTQR |r'(t)| dt ' «0 Since I is a single valued and continuous function of t, I » K'). If f(t) does not vanish anywhere, we have l'(t) •/• 0 at all the points and, then, t can be represented as a single - valued continuous function of I, t«t(l). Consequently, letting r - r(t(l)), we obtain the parametric function of the curve, with I as parameter. At each point of the curve, corresponding to a value I, the unit vector (2.4) v(l) - r'(l) Is the tangent vector, and the scalar (2.5) c(l) - |V(I)| - |r"(l)| is the curvature 2, which may be more conveniently computed In terms of the generic Indepen- 1 <' > denotes the first derivative

Analysis and Synthesis of Smooth Shapes 251 dent variable t as follows: (a.6) c - c(t) - (x' y" - x" y') / (x,z + y'2)*2 The shape of a closed plane contour is always determined by the curvature function c « c(l) (or c « c(t) and I - l(t)). Namely, the parametric function of the contour can be reconstructed as foiiows (see figure 1) (2.7) where (a.6) (a.9) I r(l) - rQ + INTQR dr dr - v(l) dl cos (theta(l)) v(l) - sin (theta(l)) I (a. 10) theta(l) » theta0 + INTGR c(m)dm 0 The curvature function c « c(t) and the Cartesian components x - x(t) and y « y(t) of the vector function r » r(t) are periodic functions and, therefore, can be expanded into Fourier series: Figure 1. Profile of the parametric equations (X, Y) and curvature (CUR) of a contour. 2 The curvature can be defined also as the inverse of the radius of the obsculating circle.

252 S. Geglio et al. N (2.11) c » c(t) - SUM (CV cosfWj t) + C2j sir^w, t)) i*1 (w, - 2 i pi / T) N (2.12) r - r(t) - SUM (A, cosfw, t) + ft sinfw, t)) 1-1 n ' ' ' where T is the period of the functions c(t), x(t), and y(t). If we denote with b « b(t) any one of these functions, Its expansion Into Fourier Series allows us to separate the components which are responsible of oscillations at low and high frequency, respectively. This Is Important because low frequency oscillations influence primarily the global structure of the contour shape, whereas high frequency oscillations are related with finer localized details. In such a way, the Fourier analysis provides us a mean to describe at different levels of detail a planar shape (see figure 2). We have two choices for the frequency analysis of a shape contour: we can perform a frequency analysis either of the vector component functions x(t) and y<t) or of the curvature function oft). However, in the case of the curvature function c(t), frequency components do not have an obvious physical interpretation, and the filtering actions do not guarantee to preserve a closed contour. It is worth noting that using the arc length I as the Independent variable has some difficulties. Mandelbrot and Steinhaus have observed that generally the arc length increases exponentially with resolution [Mandelbrot 1983]. On the other hand, vector component functions x(t) and y(t) offer great simplicity in reconstructing the filtered contour. A description of the shape at a certain level of detail can be obtained by summing the first k frequency components of the contour function b(l): Figure 2. By means of the Fourier analysis of the contour s shape can be described at different levels of detail.

Analysis and Synthesis of Smooth Shapes 253 (2.13) bf (t) - bQ (t) + b1 (t) + ... + bk (t) where bf (t) is the filtered version of the contour function and the b, (t), i - 1 k, are the first k frequency components of the contour function. The Cartesian representation of the boundary and its intrinsic representation - by means of the curvature profile - are equivalent and we have shown the way to transform one into the other; however, it is worth noting that the latter is position/rotation independent, whereas the former is not, unless we enrich the representation with (i) a non ambiguous convention about an "Intrinsic* reference frame and (ii) a specification of the translation/rotation (e.g. by means of a homogeneous matrix). 2.3. Discrete Boundary <B(DSRS)> It is the boundary of a DSRS. This is a chain of 8-connected pixels [Pavlidls 1982] which can be represented either in Cartesian terms, by means of the xy coordinates of the chain of pixels or in terms of the chain code, i.e. the sequence of steps (one of eight possible ones) between one pixel end the next one. B(DSRS) can be obtained from a DSRS by means of standard contour following algorithms. This is the starting point for applying digital processing algorithms such as the FFT (Fast Fourier Transform). However, in many practical cases, the number of pixels which comes out of the contour following process is highly redundant and it is quite useful to reduce their number by means of some method which expresses the contour as a sequence of points (rk , k «1 ,n) This corresponds approximately to sample the boundary of the SRS from which the boundary of the DSRS is generated. Different local least squares approximation procedures can be devised for this purpose. In order to apply the Fast Fourier Transform algorithm to the sampled contour, we need a number of samples which is a power of 2. Since we want to avoid a modification of the contour shape, and to preserve its continuity, it is not possible to apply the usual technique - addition of zero samples at the end. Specifically, we used B - splines interpolation and re - sampled with a power of two number. We also added a replica of samples at the beginning and the end of a period in order to overcome any problem of continuity at the extremes. We notice that interpolating the curve samples, with inter-sample distance p, implies to filter the curve with a low pass transfer function of the kind

 (a.14) (sinf)n + 1/f which has the first zero at f = 1/p (n is the interpolation order: 1, linear, 2, quadratic, etc.) [Des- slmoz 1980]. Jerry [1977] showed that the Nyquist theorem applied to curves requires that the sampling step has to be at least equal to the radius of the smallest osculating circle. Since for shapes of real objects it is meaningless to talk of the smallest osculating circle - we can go up to the molecular resolution - we consider a digitized shape as a low-pass version of the shape of the object at a given resolution. Consequently the smallest osculating circle is well defined. DSRS's - binary digital images - and B(DSRS)'s - binary digital contours - are two representations of the same thing. The former can be considered as an Implicit representation of the shape, whereas the latter as an explicit representation. Although both of them identify the same thing, they have very different computational characteristics. For example, the implicit representation is naturally suited to determine whether a given pixel is outside or inside the shape, whereas the explicit representation is the natural choice when computing local geometric characteristics. Standard algorithms are available for transforming one representation into the other: (i) the explicit representation is obtained from the implicit representation by means of contour following, and (ii) the implicit representation is obtained from the implicit representation by means of contour filling. 2.4. Boundary Filtering <LF(B)> Is the low frequency component of the boundary. <HF(B)> Is the high frequency component of the boundary. The low pass and high pass filters must be chosen in such a way that all the information is preserved: (2.15) B - LF(B) + HF(B) LF(B) conveys information about the main structure, whereas HF(B) expresses details/textures. If B is the boundary of a SRS, also LF(B) is the boundary of a SRS. HF(B),on the contrary, is a sort of closed scribble (see figure 3) centered around the origin. (High pass filtering, by definition, eliminates the DC component of a signal.) For both frequency components, there are three basic problems to be faced: (a) a representation congruent with the complementarity principle; (b) a problem of analysis, i.e. of estimating the parameters of a representation from an experimentally obtained image;

Analysis and Synthesis of Smooth Shapes Figure 3. Low-frequency (LF) and high-frequency (HF) components of a contour. A is the original shape. (c) a problem of synthesis, i.e. an interactive shape editor which allows to specify and modify the representation of a structured shape. A problem which immediately arises is the choice of cutoff frequency of the LF filter and, consequently, of the HF filter. In qualitative terms, the choice should be made in order to maximize the stability of the representation, i.e. in such a way that the structural and textural representations are as insensitive as possible to variations of the chosen frequency. Let us now discuse the criteria for choosing the filters which allow to separate global properties of the shape from finer details. First of all, each filter must give precise spatial information about the localization of particular features. This implies that the filter must have null phase, since every phase variation in the frequency domain causes a shift in the space domain. Another important requirement is that the set of filters must retain all the information of the original contour: this can be accomplished if the sum of all the filter outputs t>1, b2 , bn gives bsck the signal. Consequently, the filters have to be chosen in such a way that the sum of their amplitudes in the frequency domain is equal to unity. The above conditions can be summarized as follows: if F1(w), F2(w) Fk(w) are the frequency responses of the filters, we must have n (2.16) SUM F,(w) - 1 + JO. i-1 ' It is well known that we cannot have arbitrary high resolution in the frequency and space domain at the same time. In other words, we cannot resolve precisely a signal both in the frequency and in the space domain. A good compromise for resolution in both domains is achieved through the use of Gaussian filters. Such filters, however, pose some problem, since their responses never vanish. Another good choice is a set of sieved cosine filters, having the following frequency response:

 (cos (2 pi (log,(w) - log,,(w_)))+iya in wJ2 < w < 2w,, (2.17) F(w) - i. £ v c c 0 elsewhere where w / 2 pi is the spatial frequency; w / 2 pi is the central spatial frequency of the filter. This choice can be also motivated by the fact that the filters are Q- constant, I.e. the bandwidth is proportional to the center frequency. This property is consistent with our analysis since low resolution in frequency means high resolution in space. As a consequence, it Is possible to locate better In space finer details, corresponding to higher frequencies. Lower frequencies are responsible of global properties of the shape and, therefore, for them high resolution is space is not needed. In figure 4 the set of eleved cosine filters is shown. 3. REPRESENTING THE LF COMPONENT OF A SHAPE We represent the structure of the LF component of a shape by means of the Medial Axis Transform defined by Blum and Nagel [1978]. The medial axis, or skeleton, is defined as the set of points (internal to a shape) centers of circles which are all contained in the shape, have maximum size (let us call them m - circles) and therefore are tangent to the boundary in one or more points. The boundary is the envelope of the one - parameter family of m-circles. (See Rektoris [1989] for the definition of one - parameter families of planar curves - in our cese, the parameter is the running variable along the skeleton.) The size of the m-ciroles is a measure of the local Figure 4. Frequency responses of the filters. The first one is low pass and its output represents the lov frequency component ol the contour. The others are band pass Q - constant filters, which analyse one oc tave of the spectrum. The sum of the frequency responses, for any value of the frequency is one. There fore, all the information is preserved after filtering

 thickness of the shape and we can define a thickness function as a scalar function of the running variable along the skeleton. The skeleton of a SRS is composed of several open or closed segments which may intersect at fork points and may have open endings. Such "singular points" of the skeleton are in one-to-one correspondence with the type of tangency between the boundary and the m- circles: (a) Points of peak curvature of the convex regions of the boundary usually determine skeletal open endings, where m- circles have a radius equal to the radius of curvature of the boundary. (The exception occurs when the local thickness is smaller than the radius of curvature and in that case points of minimum curvature of the boundary do not determine singular points of the skeleton.) When points of minimum curvature of the boundary do determine open endings, the latter can be computed by tracing the normal to the boundary in the point of minimum curvature and by advancing inside by an amount equal to the radius of curvature. (b) (ii) M - circles centered on fork - points of the skeleton are tangent to the boundary in three or more points. The number of intersecting skeleton branches is equal to the number of tangent points and the unit - vectors tangent to each skeletal segment departing from a fork point bisect the angles formed between the radii of the m - circle which touch the boundary. Apart from the 'singular points" - open endings and fork points - which form e finite set, the skeleton consists of curved segments of "regular points", characterized by the fact that the associated m- circle is tangent to the boundary in two distinct points (figure 5 shows the different kinds of skeletal points). Figure 5. Different kinds of skeletal points. A is an "end point": the m-clrcle is tengent to the contour in two coincident points (which correspond to a peak in the curvature prolile). B is a 'regular point": the m-contour is tangent to the contour in two different points. C is a 'fork point": the m-circle is tangent to the contour in three different points.

 The skeleton of a shape without holes is topologically a tree where each terminal node corresponds to an open ending, each non-terminal node corresponds to a fork point, and the links correspond to the branches of the skeleton. If the shape has holes, it Is possible to have loops - closed chains of one or more links - and the topological structure is that of an oriented graph (see figure 6). The frequency content of the boundary has a clear relationship with the complexity of the skeleton-, it can be easily shown that if the boundary has n harmonics, at most the number of points of peak curvature is 2n. For each branch of the shape, the differential geometry of SRS's determines a constraint among the boundary, the skeleton, and the thickness function: (1) let s« s(t) be the natural curvilinear coordinate along the skeleton as a function of the running variable t; (2) letd*d(t)be the thickness profile, function of the same running variable; (3) let alpha be the angle between the two radii of the m - circle which join its center on the skeleton to the boundary, where the angle is measured counterclockwise from the left to the right radius (we call it aperture angle); then (3.1) cos(alpha/2) « d'/s' (See figure 7 and the appendix on the envelope of one - parameter families of functions). B Figure 6. The skeleton of a shape without holes is a tree (A). The skeleton of a shape with holes is a graph (B).

Analysis and Synthesis of Smooth Shapes 259 If the running variable t is inverted, i.e. if the skeleton is followed the other way around, the aperture angle becomes its supplement. The aperture angle is 0 or 2 pi at the open endings of the skeleton; at the fork points the sum of all the aperture angles (if all the running variables are directed to the fork point) must be 2 pi. MAT has long been considered as a useful methodological tool to explicit the structure of a shape but its practical use has been hindered by the apparent hypersensitivity to high frequency components, in which noise and/or texture are predominant. Brady and Asada [1984] pointed out the main potential defects of MAT and proposed an alternative representation for 2-D shapes, called Smoothed Local Symmetries (SLS). SLS are defined as local symmetries of boundary points plus the loci of local symmetries that are maximal with respect to forming a smooth curve and their computation results in a representation somehow different from MAT. Specifically, Brady and Asada complain that branch points do not perform reliably the task of isolating subshapes when there is a discontinuity on the bounding contour of a shape. However, these problems can be easily overcome following the frequency analysis approach: Applying MAT to the low frequency component of the boundary 1) avoids problems with boundary discontinuities, which are all contained in the high frequency component; 2) allows branch points to code reliably the information about the topology of shapes. Moreover, Brady and Asada used a rather complicate method to face the presence of noise on the contour: They chose a set of "knots" - i.e. significant contour points - approximating the boundary by means of straight lines and circles which connect the knots. Knots must correspond to significant changes in the curvature along the contour and stripping out noiee is reduced to finding significant curvature changes - a quite difficult task. In our approach - boundary filtering - noise Is automatically and easily separated from the structural information of shapes. A basic contribution of our approach is then to show how MAT can be correctly and significantly used. Figure 7. Differential geometry for a standard skeletal point "p" of the skeleton "s". "c" is the contour, "d" is the thickness, and "a" is the aperture angle. The rate of growth of the thickness is proportional to the cosine of a/2.

260 S. Gaglio et at. 3.1. Formal Definition of the Skeletal Representation We are now in a position to give a formal definition of the Skeletal Representation. < Skeletal Representation >:: = (< shape_name > < vertices > < edges >) < vertices >::»({< vertex descriptor >)) <vertex descriptor>::« <open ending>/<fork point> < open ending >::«(OPEN < ending_name > < thickness > < branch_name >) <fork point>::«(FORK <fork namexthicknessxorder> ({< branch_name > < aperture_angle >))) <edges>::«({ <branch_descriptor>}) < branch_descriptor>::«(< branch_name > < initial_vertex > < f i nal_vertex >< s keleton_segment > < thickness profile > < length > < average_thickness >) <skeleton segment>::«(parameters of a Bezier curve expressed in a local reference) <thlckness_profile>::'(parameters of a Bezier function) < shape_name >, < ending_name >, < fork_name >, < branch_name >::« ATOM < initial vertex >, < final vertex >:: = ATOM <thickness>, <average_thickness>,<length>::« positive real number <aperture_angle>::= positive real number not greater than 2 pi <order>::« positive integer number greater than 2 The Bezier polynomials [Faux and Pratt 19691 have been chosen for computational convenience, but any other parametric representation is equally acceptable. The thickness profile gives a description which is obviously independent of translation/rotation. For the skeleton_segment it is possible to use a translation/rotation independent representation, which is the curvature profile, but this is not very convenient computationally; however, it is possible to normalize a Cartesian representation of the curve by expressing the parameters - in the Bezier representation they are the guiding points of the curve - in a naturally local reference frame which we chose with the x-axis directed as the tangent unit-vector in the origin of the curve. A certain redundance is present in the skeletal representation, but this may be convenient for two types of reasons: (i) the redundance helps in defining and implementing consistency tests, (ii) it is convenient to represent explicitly certain macro - parameters, even if they can be derived from parts of the representation, when they are "meaningful" at some coarse level of shape analysis and their computetion is rather expensive. An example of the redundance of the first type is given by the fact that the specification of the <aperture_angles> in the <vertex_descriptor> must match the value of the derivative of the < thickness > profile in the corresponding < branch descriptor > and an example of the

redundance of the second type is given by the <length> of a <branch_descriptor> which otherwise would require, if not stored explicitly, to integrate the Bezier curve every time.

 4. ANALYSIS OF THE SKELETAL REPRESENTATION The Input of the analysis problem is the representation of the boundary. If we use an harmonic representation, it means that we have to specify the order N and the harmonic coefficients for the x and the y components of the contour - 4N coefficients: N (4.1) r = r(t) = SUM (A, costa t)+ B, sin(w. t)) i-1 (w, - 2 pi i) From this we can derive explicit formulae to compute the first two derivatives of the contour (r*. r") and then the tangent (v), the normal (n), the curvature (o): N r' - 2 pi SUM (-i A, sin(w, t) + i B. cos(w, t)) i-1 ' ' ' ' 2 N 2 2 r" = (2 pi)* SUM (-T A, cosfw, t) - TB.sinfw, t)) i«1 ' (4.2) v « r'/|r'| « (vx.vyl n - [-vx.vyl c - (x' y" - x" y'V|r'|3 (4.1) and (4.2) give a continuous representation of the boundary, expressed in terms of a set of periodic functions of the running variable t (normalized in order to have a unit period). It is quite apparent that, if the boundary is continuous, also the skeletal representation has to be continuous - skeleton branches and thickness profiles. However, an explicit transformation in the continuous domain is unknown and it is necessary to devise an incremental procedure which computes nearby samples of the skeleton branches and of the corresponding thickness. Afterwards, such a discrete representation can be approximated by means of suitable smooth functions. The continuous representation given by (4.1) and (4.2) is appropriate to deal with the differential geometry of the shape, i.e. with local, incremental problems, such as advancing from a regular point of the skeleton to a nearby one. On the contrary, global problems, such as the detection of fork points, require a representation which is intrinsically global. If we wish to stay in the continuous domain, one possibility consists of using implicit equations, In addition to parametric equations: (4.3) g(r) - 0 This representation would allow to perform in a straightforward way set membership classification for a given point (it Is sufficient to evaluate the function at that point and to test its sign). However, it is far from trivial to transform (4.1) into (4.3). (One might use the theorem of moments,

 applied to 2D functions, and express (4.3) as a series expansion [Budak and Fomin 1973].) A discrete alternative to (4.3) is a binary image, obtained from (4.1) by means of sampling and contour filling - in many practical cases, however, such an image is already available, as the original source of information from which (4.1) is derived. If the object has holes, the boundary representation - (4.1) or (4.3) - is split into the external contour and the contours of the holes. In such a case, we would have to add an Index to the coefficients of the representation in order to denote the individual contours. At any rate, the computational procedures that we discuss in the following are applicable to objects with or without holes if we assume that the external contour is generated in one sense (e.g. clockwise) and the holes in the opposite one. The shape analysis procedure that we implemented is based on an incremental algorithm which computes samples of the skeleton and of the associated thickness profile for a branch of the skeleton (Skeletal Branch Incremental Module), beginning and ending on singular points of the skeleton. 4.1. Skeletal Branch Incremental Module (SBIM) SBIM advances from a regular skeletal point (RSP) to another one, displaced by an assigned (small) distance deltas. A RSP can be fully described by the following attributes: (1) Cartesian coordinates of the skeletal point <rg:: «x ,y >. (2) The two boundary points, equidistant from the skeletal point and whose boundary normals meet at the same point. These points can be identified by the values of the running coordinate on the contour <tj,tr>, one for the point at the left and one for the point at the right of the skeletal tangent. If the shape has holes, we should also add the two pointers <k(,kr> to the left and right contours, respectively. In addition to the two previous attributes, it is also useful to store the two following ones, even if they are redundant, in the sense that they can be computed from the first two attributes by using (4.1) and (4.2): (3) The local thickness <d>, which can be obtained by evaluating (4.1) for tl or tr and by computing the Euclidean distance of such a point from rg. (4) The aperture angle < alpha >, which can be evaluated as follows: (4.4) cos(alpha) « -V|.vr (V| and vr may be computed using (4.2) and <. > denotes the scalar product). From a RSP descriptor , SBIM computes the local tangent <v,,> to the skeleton which, by definition, must bisect the angle between the two normals and, then, the two tangents: (4.5) v8 - ty-v^-v,! (See figure 8 for a summary of the relevant variables.)

Analysis and Synthesis of Smooth Shapes 263 Figure 8. Geometry used by the computational module which incrementally advances a regular skeletal point "re*. "C" is the contour, "d" is the thickness, "vs" is the unit-vector of the skeleton In the direction of advancement, "a" is the aperture angle, "vr" and "vi" are the unit-vectors tangent to "* contour In the two points where the m-circle touches the contour (on the right and on the ton, respectively, ol the direction of advancement). The new RSP, close to the previous one, is computed by first performing a linear estimate (this means to move from the old RSP along v$ by an amount delta.) and then by refining itera- tively the estimate. The linear estimate is performed according to the simple geometry of figure 11, which suggests to carry out the following program: 1. new,, « r + v delta. rs s s s (This is the initial estimate of the new RSP); 2. delta. = delta cos(alpha/2) (This is tha linearized displacement over the boundary.) 3. delta^ - |r'(tl)| deltab 4. delta,,. « |r'(tr)| deltab (These are the variations of the two running variables which correspond approximately to the same boundary displacement delta,,.) 5. deltad ■ sin(alpha/2) deltab (Variation of the thickness.) These computations result into two new boundary points (rb^f., rbnaht)i one at the left and the other at the right of the local tangent to the skeleton. For them to form, together with new , an actual skeletal descriptor two conditions must be met:

264 S. Gaglb etal. 1. new must be equidistant from rb)eft and rt>rjorrt. and 2. the two vectors which connect new to rt>, j. and rb, w, respectively, must be normal t(the contour. If this is not verified for the three points obtained by the linearized initial estimate, it is necessan to setup a successive approximation procedure which tries to improve the extent to which these conditions are satisfied up to a desired threshold. This is a one - degree - of - freedom problen which is solved by determining an interval of unimodality for the candidate new skeleton point moving either on one of the two normals or on a circle, of radius delta s, centered on the ok skeleton point. Figure 9 shows the sequence of RSPs found by SBIM on a branch of a shape. In addition to the previous computation, SBIM must perform a termination test, to determine whether a fork point or a terminal point has been passed (in such a case, the m-circle centerec in the new skeleton point would intersect the boundary). The test is done by using a discrete representation for the shape (the binary image of the shape) and for the m-circle (its discrete contour) and by checking that no pixel of the contour falls outside the image. In order to have e consistent formulation, it is necessary that the step size over the skeleton - branch (deltag) is i few times bigger than the sampling grid size of the binary image. The last sample found by SBIM is obviously wrong and the correct fork point can be

com puted by going back one step and by iteratively adjusting the step size. In the global computational architecture of the skeletal representation, SBIM occupies ar intermediate level, preceded by a control module which organizes the global search and followec by a module for expressing in parametric continuous form the branch functions (skeleton shape and thickness). Figure 9. Sequence of RSP's (Regular Skeletal Points) computed by the Skeletal Branch Incremental Module.

Analysis and Synthesis of Smooth Shapes 265 4.2. Control Module (CM) In order to analyse all the skeleton of a shape, it is natural to use graph exploration techniques coupled to SBIM. In particular, after SBIM reaches a fork point, it is possible to store In a stack the skeletal point descriptor (SPD) consisting of (i) the Cartesian coordinates of the point, (ii) the thickness, and (iii) the aperture angle. The exploration must start with an initial SPD and the control flow of the module is organised as follows: begin get an initial SPD & push on stack; until the stack is empty do begin pop SPD; activate SBIM; if the final point is not an ending point and it is not an already found fork point then do begin compute and push the SPD's; store the fork point; end end end The simplest way to get an initial SPD is to look for a terminal point of the skeleton because terminal points can be easily derived from points of peak curvature (either convexities of the external contour or concavities of the internal contours, if the shape has holes): at these points, the m- circles have a radius equal to the radius of curvature and they touch the contour only at one point. As a consequence, an SPD can be computed by tracing the normal to the contour at a point of peak curvature and advancing to the center of the m- circle. However, not all the SPD's computed in this way are true SPD's, because the m- circle could intersect the boundary somewhere else (see figure 10). The uncertainty can be solved by performing the same test used to detect skeletal points and by discarding the false SPD's so detected. With regard to the fork points of the skeleton, the computation of the SPD's which correspond to new branches of the skeleton is illustrated in figure 11. Furthermore, figure 12 shows an example of performance of the control module.

266 S. Gaglio et al. Figure 10. A and B are both centers ol circles tangent to the contour at a point of maximum curvature, with a radius equal to the radius ol curvature. However, A is a skeletal point, whereas B is not. Figure 11. Computation ol new skeletal branches at a fork point. SA is the ok) skeletal

branch, being computed by the incremental algorithm; CI and C3 are the two contour points at the right and at the left of the skeleton, respectively, which are tangent to the m - circle centered in the skeletal point ol the old skeletal branch. At the fork point, the algorithm finds an additional tangent point: C2. The starting description ol the two new skeletal branches can be computed as follows'. SB is directed as the bisector ol the angle between C2 and C3, and SC Is directed as the bisector of the angle between Ci and C2.

Analysis and Synthesis of Smooth Shapes 267 4.3. Branch Descriptor Estimation Module (BDEM) SBIM gives a discrete description of a branch - a list of SPO's - which is highly redundant. However, the aim of the analysis system Is to compute a concise continuous representation of the branch skeleton and of the associated thickness profile In terms of smooth parametric functions, whose order expresses their oomplexity. In other terms, we wish to transform the representation •0 s1 •■ 8m (4.8) d0 d1 ... dm alpha- alpha, ... alpha in the following representation: e - 8(u) (4.7) u-(0-1) d > D(u) where S(u) and D(u) are two smooth functions of the running variable u, normalizsd to a unit range. (S(u) Is a vector valued function, whereas D(u) is a scalar function.) S(u) and D(u) must satisfy the following boundary constraints: Figure 12. Computation of the skeletal representation. The shape on top has one external and one internal contour. The shape at the bottom has one external contour and three internal contours. The algorithm looks for end points of the skeleton which correspond to points of minimum radius of curvature (left row) and pushes the corresponding descriptors on a stack. Then it proceeds from end-points to fork-points and from fork- points to fork-points, until the stack is empty.

268 S. Gaglio et at. (4.8a) StO)-*,, D(0)-d0 D'(0) - |S'(0)| cosOJpha,, /2) (4.8b) S(l)-sm D(i)-dm D'oHS'O)! cotfalpha,,,/2) and they must comply with all the other SPD's as well as possible (while keeping the order as low as possible). In other worde for a sequence of values of the Independent variable u the following must hold: (4.9a) SQi,)? a, (4.9b) CKu,)- d, i-1.2.... m-1 (4.0c) D'(U|)S |8'(u,)| coefalpha,/2) Therefore, we have to face, for S(u) and D(u), three classic identification problems: 1. to ohoose the base functions; 2. to estimate the parameters; 3. to estimate the order. With regard to the first problem, we chose Bezler polynomials (4.10) pn (u) - SUM p. n!/ll(n-l)l u1 (1 -u)n~' n l-O ' (n +1 Is the order of the polynomial and Pl are Its guiding points)

because they are smooth and computationally convenient (In particular, they depend linearly on the parameters). For more details on Bezler polynomials and curves, refer to the book of Faux and Pratt [1979]. Figure 13 shows the approximation of a skeletal branch by means of a Bezler curve. Even If the identification problem Is non linear, It is possible to separate It into a linear part, which allows us to use standard least square techniques, and a non linear part, which can be dealt with by means of successive approximations (ese Appendix B). 5. SYNTHESIS OF THE SKELETAL REPRESENTATION The computational aspects of shapes have become, during the last years, a key point for most applications of robotics and artificial Intelligence. For thla reason, when evaluating a scheme

 Figure 13. Approximation of skeletal brsnches by means of Bezier curves. A: contour. B: skeletal branch descriptor computed by the incremental algorithm. C: Bezier guiding points. D: estimated Bezier curves (in the example, second order polynomials were found to be appropriate: note that only the guiding points of each branch need to be stored). or representing shape, Its possibility of performing well In both analysis and synthesis should be larefully taken into account; In other words, a "good" representation scheme should be powerful inough to provide, on one hand, a set of robust algorithms to recover and code the shape of ibjects, and, on the other, an interacting environment which has a clear geometric semantics fa- he users. This section shows how the synthesis of shapes can be suitably faced by means of he skeletal representation previously introduced: An interactive editing system is presented, SCULPTOR - 2, allowing the user to build 2 - D shapes and to Incrementally refine them by modi- ylng or augmenting the structure. To generate a shape by means of the Skeletal Representation Scheme we must provide the jser with primitives for dealing with the formal structure defined In the previous section. In partic- jlar, the user will have to create and edit the topology of the skeleton, the particular shape of »ach branch, and then he will have to "dresa" the structure by defining the "thickness" of each xanch. It is aleo worth noting that the shape editor can deal equally well with purely synthetic shapes or with shape representations computed from real objects: the two processes (analysis and synthesis) can then become, in this way, fully compatible. An editing session in SCULPTOR-2 can be started with a simple, single - branch shape which can be defined by the user in terms of (i) a set of guiding points for the skeletal branch, (if) a few

significant values for the thickness profile. The single - branch shape is drawn on a graphic display and its symbolic description is generated and stored. SCULPTOR-2 provides then a set of commands for performing the following operations: [a) Creation of new branches, which means to define new vertices, new skeleton segments and the associated thickness profile. (b) Modification of the shape of the skeleton of existing branches.

270 S. Gaglio et al. (c) Modification of the thickness profile of existing branches. Commands of group (a) affect the topology of a shape, whereas commands of groupe (b) and (c) have a local affect on a branch of the shape. When a command is executed, the symbolic description of the shape is updated, introducing the new vertices or modifying existing skeletal/thickness descriptors, and the new boundary Is evaluated, In order to display it and provide the visual feedback for subsequent editing commands. 5.1. Boundary Evaluation The boundary is evaluated from the skeletal representation using an algorithm which performs the opposite operation of SBIM (Skeletal Branch Incremental Module). For each branch of the representation, the algorithm proceeds Incrementally, sampling the skeleton segment and evaluating, for each sample, the tangent unit vector vg and the 8PO (8keleton Point Descriptor), which consists of (i) the skeleton sample r , (ii) the thickness d, and (lli) the aperture angle alpha. The two boundary points (rb,8ft and rbrigm) which correspond to each SPO are oomputad as follows: rb|eft * d (cos(alpha/2) vg + sin(alpha/2) vn) rbrignt - d (cos(alphB/2) v8 - sln(alpha/2) vn) where vR is the unit vector normal to vg (the geometry Is the same already Illustrated In figure 4.1). 5.2. Consistency of the Representation It is Important to point out that validity and consistency of the transformed representations are automatically taken into account by the editor, preventing the user from generating invalid shapes. The validity of the representatbn Implies several constraints on different geometric aspects, at different levels: (a) Consistency of branch descriptors. (b) Consistency at fork points. (c) Global consistency. In order to test the first type of consistency, we must consider the skeletal segment and the thickness profile of a branch. The former must not Intersect itself and this simply requires that the polygon of the guiding points Is open. The validity of the latter depends on the first and second derivative of the thickness function, which must not be too Wg. The analytical test of this condition Is rather complex and it Is more convenient to proceed by simulation.

Analysis and Synthesis of Smooth Shapes 271 Consistency at fork points Implies a constraint on the aperture angles (alpha's), as shown in figure 14. In particular, if we suppose that all branches are "entering' the fork point, then the sum of the aperture angles must be equal to 2 pi. it must also be remembered that a constraint on the aperture angle Implies a constraint on the first derivative of the thiokness function. At a fork point, other significant angles are the angles - theta's - formed between any two adjacent skeletal segments meeting at the point Also these angles must add up to 2 pi. The relation between the two sets of angles Is simple, as it Is confirmed by the figure: theta12 theta^j alpha^S + alphag/2 alphag/Z + alphag/2 thetan1 « alphan/2 + alpha.,/2 However, the opposite computation (from the theta's to the alpha's) can have pathological situations such as the case in which the fork point has three branches and two skeletal segments stay on the same line, i.e. one of the theta's is pi. This means, in particular, that when a new fork point is created, cutting an already existing branch, it Is necessary to adjust locally the previous ekeietal segment, as indicated in figure 15. The global consistency, i.e. the fact that the overall boundary does not Intersect itself, cannot be tested analytically but only experimentally, during the evaluation of the boundary. All consistency checks are built in SCULPTOR -2 and are perfectly transparent for the user, who may be warned of the type of Inconsistency and asked to re-edit the representation. Figure 14. Relationship, at a fork point, between aperture angles (alpha's) and the angles (theta's) between adjacent skeletal brsches (s1,s2,s3).

S. Gaglio et al. Figure 15, To overcome a pathology at a fork point (theta=pi) the existing skeletal branch is slightly modified; the dotted circle is the old m-circle at the fork point. In this way It is guaranteed that the representation Is transformed safely through all the editing session. In general terms, we may think of theee oonslstenoy eheoka as the 'geometric expertise" possessed by SCULPTOR-2 and placed In the library of geometric procedures. The othei main component of SCULPTOR-2 Is the symbolic manipulation part which manages the Interactive side. (Name generation, dynamic modification of structures and so on.) In future extenalora of the language, an Interface will become available with programming environments related tc action, perception or planning. Figure 18 shows three shapes generated by means o SCULPTOR-2. Figure 17 shows some Intermediate steps of the construction. Appendix C reports, after the list of all available commands, the whole editing session of figure 16. Figure 16. Three planar shapes generated by meens of SCULPTOR-2.

Analysis and Synthesis of Smooth Shapes 273 6. REPRESENTING THE HF COMPONENT OF A SHAPE A theorem by Logan [Logan 1977] defined in the field of signal theory has been used by Marr, Poggio and Ullman [1978] in order to construct a theory of early visual Information processing. The theorem demonstrates that if a one - dimensional analytic function is bandpass of bandwidth less than one octave and has no complex zeros in common with its Hilbert transform (this condition can be ignored in most practical cases), then the function Is determined by its real zero crossings, except for a multiplicative constant. According to the theory of Marr, Poggio, and Ullman, light intensity distribution in images can be represented faithfully using only the zero - crossings of the Images obtained from the original one by filtering It through a set of independent bandpass channels with one octave bandwidth. In particular, they chose a zero crossings representation of the image (called "rough primal sketch") in which discontinuities of image intensity are located by convolving the image with a bar shaped mask, which approximates the second directional derivative. Marr and Poggio [1979], In their theory of stereopsis, assume that the elements that are matched between the two images are the zero-crossings in bar-mask outputs. In both cases, zero-crossing are considered to be an expressive way to encode the shape of light intensity distribution. We have obtained a zero-crossings representation of the boundary of a planar shape. The choice of the filters of the contour functions provides us, as discussed in section 2.4, with a set of bandpass functions (bj) of bandwith less than one octave. As a consequence, the first hypothesis of Logan's theorem is satisfied and each b, can be represented by Its zero- crossings. Furthermore, since the responses of the chosen filters sum up to 1, the zero crossing representation of b:'s is a faithful representation of the contour. The location within a contour of zero crossings for a given shape is shown in figure 18. As can be observed from the figure, the components at high frequencies have the energy concentrated in particular regions, where the contour exhibits more oscillations. This fact Figure 17. Branch design in SCULPTOR-2.

274 & Gaglio et al. Figure 18. Zero Crossings Representation of the high frequency part of a contour. A: the original contour. B: location of the zero-crossings on the contour. C: samples of the High Frequency component at the

zero-crossing - zero-crossing representation. D: interpolated contour. suggests a representation of the contour In terms of the zero crossings of the curvature of Its components in which: 1. The zero - crossings of low - frequency components are spread over all the contour, since they represent wide and non-local oscillations. 2. The zero-crossings of high frequency components are concentrated In the regions in which the finer oscillations have a significant level of energy. This representation is a very effective coding of the shape of the contour since it allows a great amount of data compression snd conveys an Intuitive description of the shape at different levels of detail through the localization of meaningful points. Unfortunately, although Logan's theorem guarantees that zero crossings retain all the Information of the original signal, It does not give us any algorithm to reconstruct the signal. For the purposes of shape analysis the proposed representation Is what we need and it can be effectively used In many applications, as, for Instance, pattern recognition. The synthesis of a shape, instead, would require a reconstruction algorithm. However, it is possible to use the Information conveyed by zaro-crossing to build an approximate shape using techniques such ss fractals [Mandelbrott 1983] or polynomial Interpolation.

Analysis and Synthesis of Smooth Shapes 275 Therefore, the high frequency components of the shape can be represented (for each frequency band) by a sequence of boundary points corresponding to the zaro-crosaings of that band. Such a representation can be organized as a set of ordered lists and it can be obtained by means of the following algorithm: 1. Compute the high frequency components of rb using the set of filters presented In section 2.4: HF1(rb), HF^,... 2. For each frequency band, determine the values of the independent variable t corresponding to a zero-crossing in regions where the energy is greater than a given threshold. 3. Store the list of values of the independent variable which correspond to the zero - crossings (H.t2). The list of zero-crossings lists is (he representation of the high frequency components of the boundary. 7. FINAL REMARKS The algorithms discussed in the previous sections have been implemented in three different packages: (a) A package of data acquisition, preprocessing, filtering, contour following, etc., which is written in Fortran and runs on a DEC PDP11. (b) A package for processing the geometrical descriptors of the skeletal representation and for performing the corresponding consistency test, which is written in C and runs on a DEC VAX-750. (c) A package (SCULPTOR-2) for interactive processing the shape structures, which is written in Franz Lisp and runs in the same environment as the previous package.

276 S. Gaglio et al. APPENDIX A: ENVELOPES OF ONE-PARAMETER FAMILIES OF CURVES A one - parameter family of planar curves is defined by the function: (A.1) F(x,y,c)-0 where c is the parameter and the partial derivatives rx, Fy, Fc, Fxc, Fyg, F^, must be continue. The envelope of suoh a family can be found by eliminating c from the two equations F - 0 (A.a) Fc-° The conditions for its existence - necessary although not sufficient - are the following ones: (A.3) F^. -/- 0 Fx Fv (A.4) dot y «/- 0. xc yc The oontour of a planar form can be considered as the envelope of a one-parameter family of ciroles, where the parameter t is the running coordinate along the skeleton. The family can be expressed as follows: (A.5) F(x,y,t) - (x - x8(t))2 + (y - y8(t))2 - d(t)2 - 0 Let us now write the two equations of the envelope: (A.6) <xb - xf + (yb - y/ - d2 - 0 (A.7) (xb - x8) x'8 + (yb - yg) y's + d d' - 0 where xb and yb are the coordinates of the two boundary points determined by the envelope. The second equation can also be Interpreted as a scalar product between (rb - r8) (the vector which points from the skeleton to the boundary) and the derivative of the skeleton vector (A.8) (rb - g . r-g - - d d' However, by definition of scalar product, we may write the same equation as:

Analysis and Synthesis of Smooth Shapes 277 (A.9) d s' cos(alpha/2) « d d' because r is directed as the tangent to the skeleton, and we get the following relation which constraints boundary, skeleton, and aperture angle: (A. 10) cos(alpha/2) - d'/s'

278 S. Gaglio et al. APPENDIX B: LEAST SQUARES ESTIMATION OF BRANCH DESCRIPTORS The branch descriptors that we need to estimate are the skeletal curve and the thickness profile (section 4.3). For these functions we use a parametric representation with Bezler polynomials (4.10) and we need to fit these functions to the array of data (4.6) In the sense specified by (4.9), taking Into account the boundary conditions (4.8). It is convenient to compute first the Bezler approximation of the skeletal segment in particular, we can write the equations which express the error given by the Bezler curve for each sample of the independent variable u: ek - S(uk) - \ k»1,2,...m-1 In the second member we single out the unknown elements, that is the guiding points r|'s. (Remember that the first and the last guiding points - r0

and rn - coincide with the terminal skeleton points - sQ and sm - which are part of the data.) ek - 8UM nl/ll(n-l)l r-{ u k' (1 -uk)n_' - sk - - »o(1_uk)n + 8mukn + SUMnl/il(n-i)lr,ukl(1-uk)n"1 - s^ - n-1 » ck + SUM f| (uk) r{ i»1 The approximation errors depend linearly upon the n -1 unknown guiding points r. and therefore we can use tha classic least square method which finds the closed form solution by expressing the orthogonality between the array of errors and the array of data: (B.1) X - -(FTF)~1 FTY where T X - (r1 ,r2 ,... rn-1] unknown array Y - Ic1 ,c2 cm_ 1 lT data *™y fl<u2> r <• f1<um-1> With regard to the placement of the samples of the Independent variable u, we may Initially choose them equally spaced and then we can improve the Initial choice using a relaxation method, which can proceed as follows: For each sample of the skeletal segment, we compute which Is the point of the Bezler approximation which is closest to it, and we use the W-'n-l yu2)-,n-1(u2l parameters matrix f2<um-1>"fm-lHn-1>

corresponding values of the independent variable for running again the estimator (B.1), repeating the whole sequence, until we cannot decrease significantly the norm of the error vector. (In practice, a couple of repetitions of the procedure is usually sufficient.) With regard to the appropriate order of the Bezier curve, it is possible to proceed as follows: starting with a low order curve, the estimation procedure is repeated with increasing order until the norm of the error diverges. (In most practical cases, a value of two or three is sufficient.) The estimation of the thickness profile is similar to the identification of the skeletal segment, with the difference that we are not only constrained by the initial and the final values of the thickness, but also by the initial and the final values of the first derivative. Therefore, as regards the Bezier polynomial which should approximate the data, we know right away the first two and the last two guiding points. Therefore, for an approximating polynomial of order n, we must solve an estimation problem of order n - 4.

APPENDIX C: SCULPTOR-2 COMMANDS The user can access 3 groups of commands: A) General commands. B) Structural commands. C) Local commands. General commands are the following: A1) (STORESHAPE < filename >) > stores on a file the symbolic description of a shape; A2) (LOADSHAPE < filename >) > reads from a file a symbolic* description; A3) (STRUCTURE) > prints on TTY the actual symbolic description', A4) (RUBBER) > removes the effects of last editing operation and restores the old shape; AS) (OK) > confirms last editing operation and updates symbolic description; A6) (NOSKEL) > deletes skeleton of shape; A7) (SKEL) > redraws skeleton of shape; A6) (NAME <shapename>) > gives a name to the actual shape; A9) (BYE) > exits from SCULPTOR-a. Commands of B group allow adding branches introduolng singular points. B1) (FIRST-BRANCH) > generates a branch named 81 with ending vertices V1 and V2; the user is asked to enter: i) initial and final vertex of branch, II) order of Bezler polynomials and guiding points, ill) Initial and final thickness, iv) Initial and final angle, v) thickness values along the skeletal branch. The latter operation Is performed by means of a cursor moving along the branch. B2) (SINGULAR - POINT n < vertex-name >) > Introduces a order n singular point in <vertex-name>; the user is asked to enter for each new branch: I) position of final vertex, li) order of Bezler polynomial and guiding points, lii) thickness values (as In previous command). The new boundary is evaluated and displayed. With C group commands it is possible to modify locally a single branch and the associated thickness function. "Locally" means that they have no effect on the global structure of the shape but only on branch parameters. C1) (CHANGE-BRANCH <branch-name>) > shows the actual Bezler parameters of the branch and allows changing eome of them by means of a oursor; C2) (CHANGE-THICK <branch-name>) > shows the actual thickness values of the branch and allows changing some of them by means of a cursor.

Analysis and Synthesis of Smooth Shapes 281 Editing session The following interaction is the whole editing session to generate the central shape of figure 16. SCULPTOR - 2 TYPE (help) FOR LIST OF COMMANDS > > (FIRST_BRANCH) /* building the body '/ # initial vertex? [x y] 243 414 # final vertex? [x y] 243 312 # bezier order? 1 /' no guiding points are asked for '/ # ok? [y/n] /" order 1 (straight line) polynomial "/ y # initial thickness? 52. # final thickness? 44. # initial angle? [in grades| 60. # final angle? [in grades] 110. # enter thickness values using cursor # f«forward, b« backward, e = value, q = quit # 100 samples available q /* no intermediate vaiues wanted */ # sampling step? 0.01 /* 100 skeletal samples — > 200 boundary points */ /* draws body */ > > (OK) /* confirming operation and updating description */ > > (SINQULAR_POINT 3 V2) /" building legs */ # new branch n. 1 # final vertex? [x y] 123 34 # bezier order? 6 # distance? /* guiding points are entered as */ -35. /* distances from the segment "/ # distance? /* joining initial and final vertex; */ -20. /* positive and negative numbers are "I # distance? /* used to indicate the two sides of */ 20. /* the segment. */ # distance? 30. # distance? 50. # ok? [y/n] /* shows guiding points on display */ y /* draws skeletal branch */ # final angle? [in grades] 0. /* to have a closed shape '/ # final thickness?

282 S. Gaglio et al. 2. # enter thickness values using cursor # f 'forward, b«backward, e«value, q«quit # 100 sample available /* cursor appears at the beginning of the branch */ 331 /* cursor moves fwd of 33 samples */ e — > thickness value? 15. 33f e — > thickness value? 10. 10f e — > thickness value? 5. q # new branch n. 2 /* see fig. 17c */ # final vertex? [x y] 363 34 # bezler order? 6 # distance? 35. # distance? 20. # distance? -20. # distance? -30. # distance? -50. # ok? [y/n] y # final angle? [In grades] 0. # final thickness? 2. # enter thickness values by means of cursor # f-forward, b« backward, e«value, q«quit # 100 sample available 331 e — > thickness value? 15. 331 e — > thickness value? 10. 10(e — > thickness value? 5. q # sampling step? 0.005 /' 200 skeletal samples - - > 400 boundary points */ /" now evaluates the new boundary and draws it */ >> (OK) > > (SINQULAR_POINT 3 V2) /• building arms, same as In figs. I7a,b */

Analysis and Synthesis of Smooth Shapes # new branch n. 1 # final vertex? [x y] 23 437 # bezier order? 5 # distance? -35. # distance? -20. # distance? -10. # distance? 20. # ok? [y/n] y # final angle? [in grades] 0. # final thickness? 2. # enter thickness values using cursor # I = forward, b« backward, e = value, q-quit #100 samples available 33f 5b e — > thickness value? 15. 251 e — > thickness value? 10. q # new branch n. 2 # final vertex? [x y] 414 437 # bezier order? 5 # distance? 5. # distance? 5. # distance? -10. # distance? -10. # ok? [y/n] y # final angle? [In grades] 0. # final thickness? 2. # enter thickness values using cursor # f« forward, b = backward, e = value, q = quit # 100 samples available 30f 31 e — > thickness value? 15. 401 e — - > thickness value?

284 S. Gaglio et at. 5. q # sampling step? 0.005 /* evaluates the new boundary and draws it */ > > (OK) > > (NAME mani) /' gives a name to the shape */ > > (STORESHAPE mani.dat) /* writes skeletal repr. on a file */ > > (BYE)

PART III Chapter 2 SHAPE AND MOVEMENT Raffaele Boccardo, Retro Morasso, Carmelina Ruggiero Giulio Sandini, Vincenzo Tagliasoo, Paolo

Vacchina CONTENTS 1. Introduction 2. Three Dimensional Active Vision 2.1 Building the volumetric sketch (occluding contours) 2.2 Refining the volumetric sketch (stereo images) 3. Representing the Surfaces of Objects 3.1 Tomographic Representation 3.2 Polygon Mesh Representation 3.3 Primitive Instancing of Analytic Representations 3.4 Composite Surface Representation 3.5 Local Surface Geometry 4. Constructive Shape Assembly 4.1 Geometric Modeling Systems and Solid Modeling 4.2 Constructive Solid Geometry (CSG) 4.3 An Example of CSG: PADL-2

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasco (Editors) © Elsevier Science Publishers B.V. (North-Holland), 1986 287
SHAPE AND MOVEMENT Raffaele Boccardo, Pletro Morasso, Carmelina Rugglero Giulfo Sandini, Vincenzo Tagliasco. Paolo Vaccnfna The complementarity of movement and shape is analyzed with the help of a Sculptoring Metaphor. In other words, representing a shape is viewed as analogous to the performance of a sculptor who produces a shape by carving fragments out of a solid block. In particular, this metaphor is pursued in considering Active Vision, i.e. the process of building the 3-D model of an object by means of multiple images obtained by a viewer who moves around the object. Finally, two other types of shape representation are considered, surface representation and volumetric representation, reviewing current techniques (tomographic representations, sculptured surfaces, and constructive solid geometry) from the point of view of the shape - from - movement and movement - from - shape complementarity. 1. INTRODUCTION Movement generates shape and shape generates movement. The meaning of this statement, i.e. the fundamental complementarity between movement and shape, can be expressed in a variety of fields, ranging from the analysis of human skills, to robotics, and to the visual arts. In the latter case, for example, the modern art critique has developed the ooncept of works of art as complex semantic structures [Mukarovsky 1944], expression of the activity and construction process of spatio-temporal shapes [Ragghianti 1946]. In other words, the artistic analysis of an object implies the same emphasis on a language of pure form, independent of the utilitarian flavor of verbal language1, which is typical of modern techniques of computational geometry in CAD, robotics, and computer graphics. Shape may be produced by movement in many different

ways: carving, molding, painting, assembling, knotting, gesturing, etc. When carving or sculptoring, for example, the sculptor starts with a block of some kind of material and chops off, with his chisel, pieces and fragments: bigger ' Let us consider an example. For verbal language (and also lor common sense reasoning) a chair is conceded as an Instrument lor sitting and Its shape is relevant only from the point ol view 01 sitting: dimension, robustness, material, etc. For the artistic analysis a chair Is a structure, in which the relation ot each element to anyone else must be equally accessible, although not necessarily explicitly expressed.

 at first, and then smaller and smaller. The process can be expressed simply, by means of set operators which operate on three-dimensional sets of points. Namely, starting from an initial_block. shape < - Initial block the process proceeds iteratively as follows shape < - shape - fragment^ which can be summarized by shape « initial block - Union^fragmeny where ' -' and 'union' are the usual set operators (with some attention to the regularity of the sets [Requicha 1980]). The fragments are specific sets of points which are characterized both by their shape and by their location (position and orientation with respect to the initial block). They are the materialization of constructive movements: the strokes of the chisel. The sculptoring metaphor is computationally useful only if it is possible to give a geometric characterization of the constructive movements of a real or virtual chisel. A particular example, that we shall consider in a following section, is a formulation of three-dimensional active vision, i.e. the process of Integrating visual information about the shape of an object while someone is moving around it or while he is turning it In his hands: the metaphoric chisel, in this case is light (or a laser beam, If you like). The sculptoring metaphor is not appropriate for modeling a similar phenomenon: the shape formation processes due to natural erosion. Here the natural chisel works with infinite strokes, and with an uncountable number of chisel sizes (from large ones to infinitely small ones); this phenomenon determines the recursive nature of natural shapas which is well captured by the formalization of Fractal Geometry [Mandelbrot 1963]. The difference between sculpture and erosion also has a general meaning: it Is the difference between the cybernetic (or A.I.) point of view and the physic point of view, a difference which was already discussed In the prolegomena. However, the difference is somewhat fuzzy because the same shape could be the result of either and it is easily overcome in an artistic framework (think, for example, of the Japanese rock gardens where naturally eroded rocks are arranged as man made statues). Molding Is another paradigm of shape formation in which movement plays a key role. Whatever material Is used (from clay to plasticine) the volume of the shapa remains approximately constant when molding It and the effect of the shaping movements Is to create branches, swellings, recesses and to deform them, a process which resembles the computational formulation, based on the notion of skeleton, which is described in the previous chapter. Assembling shapes is a method of generating a shape from simpler shapes, by means of "assembly movements". This is the domain of Constructive Solid Geometry (CSQ) that we briefly discuss in one of the following sections. The building blocks of an assembly have a standardized shape and the building process is characterized by movements and by set operations, in a similar way to the sculptoring metaphor. However, there is a significant difference: the "fragment" of

 the sculptoring process disappears after each "chisel stroke" and what remains is the sculptored shape, which is an analogical representation: the features of the shape can be "measured" from its representation. On the contrary, the CSQ representation is a symbolic representation, structurally a tree where all the composing elements preserve their full individuality: the features of the shape, in this case, are "inferred" from the features of the composing elements, which must be explicitly known or readily computable. Painting and Gesturing are two other modalities that people can use in order to generate shapes by means of movement. In the latter case, the kinematic structures of trajectory formation that we discussed in the first chapter of this book are the neuromotor constraints from which highly articulated systems of trajectories can be built: for example, the repertoire of mimic expressions codified in the theater performance [Rosa 1929] or the sign language for deaf people [Poizner, Newkirk and Bellugi 1983). Painting and brush-work, particularly in the case of Chinese calligraphy [Chiang 1973, Kao et al 1984] which uses very soft brushes, are paradigms of shape formation which materialize the skeletal representation discussed in the previous chapter: the trajectory of the brush-tip is the "medial axis" and the pressure on the brush is transformed into the width of the brush tip (pressure modulation along the "medial axis" is the shaping mechanism). Summing up, the different modalities of shape generation lead to three different classes of representation: - sweeping representation, - analogic spatial representation, - symbolic spatial representation. The skeletal representation falls in the first category, the sculptoring metaphor relates to the second, and the assembly paradigm relates to the third. Shape generates movement in a reciprocal way: aiming, targeting, placing, facing, inserting, grasping, catching, touching etc. are all movement paradigms which are driven by the shape of objects. For example, aiming or targeting, that we studied in previous chapters, are basic movement paradigms whose argument, the target position, is usually expressed as a polnt_of_a shape instead of an absolute point in space. Placing requires one more step than targeting, i.e. a soft impact and an adjustment (of the hand, of the foot, or of some tool) to establish a firm and stable contact with the surface of an object. After placing, it may be necessary to slide along the surface while keeping the contact and a given level of pressure. In general, the shape of objects drives the generation of movements by means of intermediate visual or tactile information which are more closely related to the surface of the objects than to their volumetric structure. In the tactile case, in particular, compliance is an important and complementary aspect because it specifies some of the attributes of the interface between the actor and the environment. The superficial and volumetric representations of an object, although complementary, have a very different meaning from the point of view of manipulation. Namely, the representation of

 the surface implies an attitude to navigation which may start visually but is ultimately haptic (tactile + proprioceptive) [Allen and Bajcsy 1985]. The skill of haptic navigation (similarly to any other navigation paradigm) is based on an appropriate system of coordinates and on a few navigation tools, which may allow to estimate surface features such as valleys, ridges, corners, passes, plains and to solve navigation problems such as descent in the steepest direction, edge following, etc. Complementarity of representations is a basic issue for perception too [Besl and Jain 1985], because recognition implies awareness of something already known and therefore a vision system cannot understand what it cannot represent internally. In general, the concept of shape is an internal object model constrained by input (perception) and output (manipulation) requirements. The various schemes of representation can be valued against a number of formal properties, such as consistency, uniqueness, and completeness. A consistent representation is a representation which corresponds to a possible well formed object and it would be desirable that the operators which modify a representation have built-in consistency checks. A representation scheme is unique if two different representations always correspond to two different objects. In the opposite case (which is quite common) it may be necessary to use equivalence operators that test whether or not two different representations correspond to the same object. Completeness, finally, refers to the domain of a representation, i.e. to the class of objects that can be represented within the scheme of representation. 2. THREE-DIMENSIONAL ACTIVE VISION Notwithstanding the strong emphasis put by Gibson and others on the ecological nature of perception [Gibson 1979], the point of view which is assumed implicitly by most of those working in natural and artificial vision Is still image-centered. As a consequence, vision is conceived as some sort of surface formation process, in which image patches are first Isolated, using linear or regional segmentation techniques, and then the patches are "rounded" and "shaped" taking into account shading, stereo or disparity, optic flow etc. The result is what Marr called a 2.5 sketch of the scene depicted by the original image [Marr 1982). The image - centered and data-driven nature of this process was expressed explicitly by Barrow and Tenenbaum [1961] when they described vision as a hierarchy of computational layers "registered" in the image domain. The alternative approach consists of taking an object - centered point of view. In this case, the leading role is played by a solid image of the world and not by the retinal images: the latter are ephemeral representations whose purpose is to help building the solid image. From this point of view, vision is such a constructive process. The fundamental role of 3D representations is supported by the experiments by Shepard, concerning mental spatial rotations [Shepard 1982], and it can be appreciated when we think of

 vision as an active process that requires the Integration of different viewpoints. For an object- centered approach multiple views are the natural requirement for building the master representation, whereas for an image-centered approach they are a source of enormous complications. Among the different methods for building solid models of objects from multiple views we shall consider two: One uses occluding contours and the other uses stereo image pairs. The occluding contour of an object, from a specific viewpoint, is a closed contour in the corresponding image which separates the figure from the ground. The appropriateness of occluding contours for building 3D visual interpretations, in absence of any other 3D cue, is supported by the experiments on motion perception by

Johansson, who reported the compelling illusion of 3D motion which Is generated by displaying moving flat shapes on a uniform background [Johansson 1975). The occluding contours of objects can be estimated in different ways: in some cases simple thresholding of the Image Intensity is sufficient but In others more complex computations are necessary. In particular, for a moving actor a very powerful and natural method consists of estimating the moving edges from image sequences [Neuman 1980, Bridwell and Huang 1983, Jain 1983, Lawton 1983). Baumgart [1974], Martin and Aggarwal [1983], Massone et al. [1985], Morasso [1985] investigated how a volumetric representation of an object can be generated by means of multiple occluding contours. Multiple views can be obtained either by means of multiple cameras which look simultaneously at the same object or by means of one camera which "navigates" around the object (in the same "ecological" way in which humans look at objects, never statically but with a continuous active process of vlsuo - motor exploration). A synergic contribution to build 3D representations of objects comes from the processing of 3D binocular stereo Images which produce depth Information. Occluding contours and stereo images provide complementary information: an ocoluding contour constrains the object Inside a conic surface7 and the corresponding stereo images specify a bas relief inside the oone3. In both cases, the 3D representation of the object may consist of a 3D {holistic) memory, a generalization of the 2D memory which is used to store images. The 3D memory plays the key role of accumulating and integrating the information which comes from different views without requiring to recognize each of them. It Is quite clear, Indeed, that it Is much more convenient to apply cognitive processes starting from a 3D representation than to use cognitive processes to infer a 3D representation. Moreover, on this memory it is possible to perform measurements whloh provide movement planning with geometric Information. A 3D holistic memory can be implemented in different ways. One possibility is to use a tasselation of space with voxels (volume elements) of uniform size [Goldwasser 1984], another possibility, which Is more efficient specially as regards memory requirements, is to use octrees [Jackins and Tanimoto 1980, Meagher 1981] or other methods which perform hierarchical decompositions of space with a variable resolution [Srihari 1981]. 2 The spatial location ol the vertex of the conk: surface is the same location of the eye. The conic surface can be regarded as the sat ol straight tries which Join the vertex to the points ol the occluding contour. 3 The "bas relief which corresponds to a depth map Is the difference between the cone of Infinite length which ie generated by the occluding contour and a bundle of conic or pyramidal fragments whose lengths are modulated by the depth map.

292 R. Boccardo et al. in the following we discuss some details of specific algorithms that we used in order to gain some preliminary experience about implementing these computations. 2.1. Building the Volumetric Sketch from Occluding Contours We do not consider here the problem of extracting the occluding contour in the image of an object in the general case. We simply assume that this can be done using standard image processing procedures (in the example of figure 1 we used well contrasted Images for which simple thresholding of the intensity level gives a satisfactory performance). A ray in the scene that converges at the eye location corresponds to any point of an occluding contour: the ensemble of rays envelopes a conic surface that contains the object. In order to give coherence to the constructive process of the object representation, it is necessary that all the geometric entities are referred to a common system of coordinates (the object reference system). Therefore, the first operation for any specific point of view is to find the location of the eye4. From this, the spatial equations of the rays can be computed in two ways which are described in the following. (a) One method is the classical photogrammetric method that requires to estimate the camera parameters (position/orientation of the image plane with respect to the environment, position of the center of projection with respect to the image plane, position/orientation of the image window with respect to the image plane, compensation coefficients of the image distortions due to non linearities of the optics and of the sensor geometry); when this is known it is possible to transform the image coordinates of the points of an occluding contour into the corresponding three-dimensional coordinates and then It is possible to compute the spatial parameters of the rays. (b) Another method consists of using what we know of the geometry about the background, exploiting it as a calibration structure. Any point of an occluding contour is also indeed a point of the background. Therefore, if we are able to know the spatial location of the points of the background we can bypass the camera calibration step: we compute the spatial parameters of the rays of an occluding contour simply by joining the spatial location of the eye to the spatial locations of the points of the background. In the example of figure 1 a known orthogonal grid is placed in the background which allows to interpolate easily the spatial location of all the

points of the image, considered as point of the background. The latter method is more robust than the former one because It is insensitive to distortions of the images, but it requires to have a suitable background. Therefore, in general cases (as in the case of human vision) both methods may be used. 4 The information about the position and orientation In space ol the eye is ot haptic nature, at least In part. There- lore, the 30 interpretation ol stereo images is a nioe example of the deep interdependence among the different sensory modalities.

Shape and Movement «. \ Figure 1. Multiple views of an object against a calibrated background. The second step in the construction of the solid model is the initial estimate of a volume (the marble block from which to "carve" the model). The size and locetion of this block can be estimated in different ways. For example. If we use the limit rectangles of the occluding contours from multiple views, it is easy to find a prism which contains the object (see figure 2). But other sources of information can be used (e.g. haptic information or cognitive information, i.e. expectations about the object) Improving the initial approximation of the block. %4 . Figure 2. Initial estimate of the block containing the object.

294 ft Boccardo et al. After the Initial block Is available, the carving • • - - can be organized in two different ways. (a) Solid to image. It consists of scanning the block voxel by voxel and of projecting each voxel toward each viewpoint, computing the Image coordinates of the projection rays. If some of the pixels fall outside the *• - ending occluding contour, then the voxel Is deleted from the solid memory. The • • ■ \s cumulative and it allows to carve more and more fragments as more and more views are available. Different scanning methods of the solid can be used, from simple sequential scanning at fixed resolution to recursive scanning which • •»::•: from low to high resolution (according to the octree concept). (b) Image to solid. It consists of Isolating convex patches of the Im • outside the occluding contours and of projecting them In space (the projections have a pyramidal shape, with the pyramid vertex in the eye location). The pyramids are then carved out, computing their I : • with the current solid representation and deleting it. Different variations of the algorithm may be conceived, according to the method which is used to compute the intersection. Figure 3 shows a volumetric re* ^ ntation of the human body obtained by means of nine views. Figure 3. Volumetric (voxel by voxel) : • =;

= tatton of the object.

Shape and Movement 295 2.2. Refining the Volumetric Sketch by means of Stereo Images In binocular stereo vision (i.e. when the same object is viewed by two eyes) It ia possible to estimate a "bas relief of the object (i.e. the shape of the part of the surface which ia in view) by measuring differences among the two images [Qrimson 1981]. Specifically, it Is necessary to measure, for many "elements" of an image6, the disparity from the corresponding point of the other image (i.e. the angular difference of an image element in the two eyes, as it is shown In fig. 4): the result of this computation is a disparity map from which it is possible to compute a depth map by means os standard trigonometric calculations. The difficulty of the task is basically determined by the correspondence problem: to identify which point of an image corresponds to a carefully selected point in the other image. This problem Is difficult because the estimate Is extremely sensitive to noise. The approach proposed by Nishihara [1984] overcomee this difficulty at the expenses of resolution, by matching image patches of band-pass filtered images (at low, medium, and high resolution) according to a hierarchical method (i.e. from the iow resolution to the high resolution). This algorithm, whioh uses a correlation between corresponding image patches, solves the correspondence problem in an indirect way and provides rough but robust range information. The modification of this algorithm, that we developed [Morasso and Sandini 1985] uses a pyramidal structure which improves the efficiency of the algorithm and guarantees a speed inversely proportional to the resolution necessary for a given task. The process, starting with a stereo pair of images (figure 5), is organized Into the following steps: Figure 4. Disparity in stereo vision. Disparity is defined as the angular difference in position of the image elements in the two eyes which correspond to the same object point. From disparity it is possible to recover "depth". 5 The elements that are matched may be quite different. We may consider all the points of the images. However, since this results in a very noisy estimeta, It Is better to usa more structured elements, such as edges, corners, patches and other geometric features of the original images or of band-filtered images.

296 ft Boccardo et al. Figure 5. Stereo pair of images from one view. (a) Multiple resolution Laplacian Filtering. Both images are band-pass filtered (with a Lapla- cian of Gaussian operator) at three different acales of

resolution, i.e. using convolution masks of different sizes (small, medium, large). The output is a set of three filtered images (figure 6) for each Image of the stereo pair. The computation is organized according to a pyramidal scheme: the different filters operate on images whose (virtual) size is inversely proportional to the (real) size of the convolution mask and this allows to use a constant (virtual) size of the convolution mask (the smallest one) and to store smaller images for the lower resolution filters6. (b) Image Clipping. The filtered images are clipped to produce a ternary :. : • = tation: -1 for the pixels which are negative (below a small dead zone, which takes care of the noise), 0 for the pixels inside the dead zone, and +1 for the pixels which are positive (above the dead zone). (c) Image Tasselation. The clipped images are subdivided into patches with a patch size (we used 16x16) roughly equal to the size of the virtual convolution mask (11x11). We get 4x4 patches for the low resolution, 8x8 for the medium resolution and 16x16 for the high resolution. (d) Disparity Map. The disparity map is estimated by means of a method which uses a cross correlation technique and exploits the multiple resolution. The method starts with the low resolution tasselated pair of images. For each patch of one image, the algorithm computes the cross correlation with the other image and it looks for the horizontal displacement which gives the maximum correlation (the interval of search Is twice the size of the patch and is symmetric = •; - the zero displacement): the result is a 4x4 array of disparity numbers, with a special code to Indicate that no reliable correlation peak was found. This disparity 9 The original Image6 ol the examples have a resolution ot 256x256 pixels. The high Irequency liner uses a nxii convolution mask which produces a 250x256 tillered image. The medium irequency lilter operates on an image with a virtual size which is hall the real sue (128x128 it is obtained liom the source image by means ol local averaging) since the real sue ot ihe medium resoiuiion mask is twice that ol ihe high resolution lilter. then il it possible to use ihe seme 11« TI virtual mask. In a simiter way. ihe low resolution liner operates with a 84x84 image and ihe same virtual convolution masx

Shape and Movement 297 EJ- I Figure 6. Multiple resolution Laplacian filtering. map is used to guide the computation at the medium scale of resolution in the following way: since esch patch at low resolution is split into four patches at medium resolution, for each of them the cross correlation is started with a central displacement equal to that found at low resolution or it is not computed at all if the computation at low level was not satisfactory. The result of the second stage is then an 8x8 disparity map. The third stage proceeds in a similar way, producing the final 16x16 disparity map (figure 7 shows this map, together with the clipped tasselated images). From the disparity map it is possible to compute the depth map7. The resolution in depth near the fixation point is a function of the camera parameters which can be adjusted to achieve the desired precision. In the example presented above, we used a lens with a focal length of 50 mm; the distance between the cameras was 23 cm and the fixation point was about 1.5 m from the cameras. The resulting depth resolution was less than 2 cm with a 256x256 stereo pair. The depth map can be ueed to carve voxels out of a solid array which was Initially filled. Concepts and remarks that we made about this aspect in the previous section can be repeated here. Figure 8 shows the solid representation which is obtained using, in addition to the image pair of figure 5, another pair from a view at about 90 degrees from the former. Solid models of objects can be obtained from depth maps by means of octrees, according to a method proposed by Connolly [1985]: a range image is initially converted into a quadtree format, the quadtree is extended into an octree, and finally octree descriptions from different views are rm • - • in a common representation. Summing up. the three tillered images have resolutions 256. 12B. 64. 7 Depth maps - also known as range images - can also be obtained by means ot active methods, that is methods which use a controlled illumination to find depth directly, without the processing ot Intensity images. Laser range finders use coherent light to measure either the lima_ot_tlight jSvetkofl et al. 1984] or to solve a triangulatjon problem [Pipitone and Marshal 1083). Also noncoherent light can be used to solve the latter problem by projecting light patterns - spots, lines, stripes, or grids - on the object (Hall et al. 1982). >| I r I I,"

 •m Figure 7. Disparity map and clipped tasselated Image. Figure 8. Volumetric representation (voxel by voxel) from two views. The • •»- of solid construction which is driven by sequences of occluding contours operates In a longitudinal way (i.e. It carves out fragments on the sides of the viewed object). A drawback of this feature is that the shape formation • >-- ■ ■ is blind to concavities. The contribution given by the depth information extracted from stereo pairs of I • ^ Is perfectly complementary: the method operates in a - way (I.e. It - ■ out fragments in the front of the viewed object) and it can work equally well with concave and convex structures.

Shape and Movement 299 3. REPRESENTING THE SURFACES OF OBJECTS Representing an object as a collection of lines is often inadequate, since surfaces, not lines are necessary in the majority of cases when working with 3D shapes: this is the case, for example, in hidden surface removal for realistic display, weight or volume calculations, 3D representations of scenes for computer vision. Surface information is inherently three-dimensional, therefore graphic processing is necessary to display it on a screen. From the application point of view, the need to represent 3D shapes usually arises in two types of situations: representation of "existing1' objects (i.e. building object models from experimental data) and representation of "non existing" objects (I.e. building object models from descriptions, requirements, ideas, etc. as in computer aided design). 3.1. Tomographic Representation As regards the representation of existing objects, a typical example of great interest is visualizing human body organs, an achievement made possible by the advent of x-ray computerized tomography that took place during the last decade and that is being followed by the evolution of other "imaging" modalities [Hermann 1960]. The output of the imaging systems consists of a set of two-dimensional digital images of cross sections of the body with a sequence of parallel intersecting planes. The set of images ("slices" of the body) are the raw experimental data which need to be processed in order to obtain meaningful information. Although the techniques that have been developed come from the biomedical field, a slice by slice representation of objects can also be regarded as a convenient starting point in a variety of non medical problems. "Slicing" is certainly a very natural human experience and many man-made objects or environments clearly indicate a slice - by - slice ordering or decomposition. Different types of processing can be applied to the raw data. Reorganization techniques rearrange the set of 2D images achieving a different set of 2D images [Udupa 1983]. Since both input and output are two-dimensional, no particular graphic processing is required. The techniques used are, mainly, re-slicing and re - projection. Re-slicing produces, starting from a given set of slices, another set of parallel intersecting planes with arbitrary orientation. Re- projection reverses the computerized tomography process, projecting the tomograms on a plane: the projections at different orientations of the plane may be displayed in sequence. Re - slicing and re - projection are useful not only from the point of view of display but also from the point of view of the representation: it is important indeed to have a "natural" alignment of the tomographic axis with some significant axis of the object.

300 R. Boccardo et al. Surface representation techniques aim to make explicit the ap- ^ : ce of the object eurface. This requires a significant step from the pure re - arrangement of the raw images: the step consists of separating the surface points from the external points and of ordering them in a way that makes explicit neighborhood relationships. Detection of surface points requires to apply image pr« = ing techniques to the slices. For well contrasted images, however, simple thresholding of the image Intensity Is sufficient to transform slices into binary images (figure 9) and then standard contour following techniques [Paviidis 1982] provide the contours of the slices as dosed chains of points (figure 10). Moreover, there are methods that provide directly surface points: one, for example, uses a laser range finder [Faugeras et al 1984] together with a scanning table that holds the object and Is moved according to a cylindrical coordinate system. An analogy can also be drawn • = : :n Figure 9. Clipped Images of selected slices of the tomogram of a statue head.

Shape and Movement 301 Figure 10. 3D visualization of the contours of the tomogram of a statue head.

302 R. Boccardo et al. this type of object modeling and the "integration" of haptic information: this consists Indeed of the tactile detection and the spatial reconstruction of eets of points In space. a In any case, the result Is a set of points in R defined by their coordinates (x,y,z) in an object centered coordinate system. This is the initial = • : -ntation, where neighborhood relationships between points are not explicitly represented. The next step is introducing these relationships in the = • --ntation. A natural idea is to look for a polyhedron whose ■■ ►are the measured points. The polyhedron is not unique in general but = .- techniques are available which constrain the search in a suitable way [Keppel 1975, O'Rourke 1981, Faugeras et al 1964]. The result is a trlangulation such as that shown in figure 11. Trlangulation is a particular case of covering a surface with polygon meshes. 3.2. Polygon Mesh R • -■-ntation Polygon meshes are collections of vertices, :••:-, and polygons and can be : • =; = nted In : ways whose advantages and disadvan ■■•■■: can be evaluated In terms of storage requirement, ease of : = sing or identifying shape elements, display efficiency, error checking etc. Each polygon may be = • -nted by a list of vertex coordinates stored In the order in which they would be encountered when traveling around the polygon. For a polygon mesh, however,

the need eriees to avoid duplicating the coordinates of shared vertices and to achieve an explicit representation of shared >< and -.So, different representations have been sought. One : • ::: tation defines polygons by lists of pointers into a vertex list, in which each vertex of the mesh Is stored just once. This representation is memory efficient and allows easy ■» Figure 11. Triangulation of the tomogram of a statue head.

 editing of the vertices, however finding polygons that share one edge Is not easy, and shared edges are drawn twice when displaying the mesh. These problems can be solved representing edges explicitly. An edge list ia used and polygons are represented as lists of pointers to it. Each edge of the edge list points to a couple of vertices in the vertex list and to one or two polygons to which it belongs. When displaying the mesh, edges rather polygons are accessed, avoiding multiple display of shared edges. Since meshes are often generated interactively, consistency checks are important: to check that all the polygons are closed, ali edges are used at least once, all vertices are referenced by at least two edges, all edges are part of at least one but no more than a maximum number of polygons, each polygon has at least one shared edge etc. Polygon meshes can be used on their own or as a preliminary phase of some surface Interpolation method. Polygons can model any object. However, If the object Is curved, a very large number of polygons may be required. For this reason algebraic surfaces and surface patches are often used. 3.3. Primitive Instancing of Analytic Representations A general surface S In thm dimensions can be described by means of implicit equations in the following way S » {r: f(r;p) * 0} where r is a point in E3 and p is a parameters vector. f() identifies a family of primitive shapes and instancing one means to pick up a value of p and to look for the points which satisfy the implicit equation f«0. For example, planes and quadrice are described by algebraic equations of the type a'r + b « 0 r'C r + d'r + e - 0 where a,b,C,d,e are arrays of parameters (C is a matrix and e Is a scalar). Quadric surfaces may be spheres, ellipsoids, cones, cylinders. This family of surfaces has been extended by the definition of superquadrics [Barr 1981]: they generalize the basic quadric surfaces by producing a continuum of forms with rounded edges and filleted faces. The standard alternative approach is to use an explicit parametric surface representation: S « {r: r-r(u,t;p)}

 where the Cartesian components of the vector r are scalar functions of two common Independent variables - the running parameters u,t. The two types of representation have different computational characteristics. Generating a surface, fa example, la easier in the case of the parametric representation because it is sufficient to calculate the values of the Cartesian oomponents for different values of the running parameters. However, the implioit representation ia better when performing set membership classification, that is when testing whether a candidate point Is on the surface or outside it. An interesting parametric representation is the spherical harmonic representation [Schudy and Ballard 1976, Ballard and Brown 1982] which oonsists of expressing - with a Fourier series expansion - the radius from a focal point as a function of spherical angular coordinates: The running parameters are the two angles and the shape parameters are the Fourier coefficients. Although the representation is useful only for convex objects and for a restricted class of non oonvex objects, we deem it possible and meaningful to try to extend it to more complex object topologies. Primitive instancing is very compact and very effloient from many viewpoints but It Is not suitable to express arbitrarily complex shapes, In which It is natural to use some kind of composition mechanism. 3.4. Composite Surface Representation A smooth surface representation of complex shapes which cannot be captured by a single formula consists of blending together a mosaic of patches. Surfaces are constructed plecewise, sewing the patches together with specified continuity conditions (they are also called sculptured surfaces). The patches are usually generated by blvarlate polynomial functions of the type (1) r - r(u,t) where u and t are the two free parameters: (1) are also called parametric equations of the surface patch. These equations induce naturally a system of coordinates on the patch, as It Is shown in figure 12. Many types of bivariate patches have been proposed: Coons' patches [Coon 1964], Bezier's patches [Bezier 1972], bicubic spline patches [Gordon and Riesenfeld 1974], eto. In particular, bicubic spline surfaces are defined as follows: N M (2) r « r(t,u) - Sum Sum r,, B,(t) B,(u) l-O j-0 " ' ' where Bk's are cubic B-spline functions8 and r,,'s are the vertioes of a characteristic polyhedron

Shape and Movement 305 Figure 12. Bivariate parametric patch. (they are also called "control points"). For any value of u, (2) defines a spline curve in space function of t; similarly, for any value of t, (2) defines another spline curve function of u. The control points Influence the ahape of the surface indirectly: the surface Is closest to any of them (say r,,) when the corresponding B- spline functions (B,(t) and B.(u)) are closest to their peak value. Figure 13-left shows the control polyhedron of a surface, figure 13-center shows some of the spline curves that cover the surface, and figure 13-right shows a shaded Image of the surface. For surface formation, similarly to trajectory formation, we may define a direct problem and an inverse problem. The direct problem consists of generating the surface when the control points are given. This problem is solved by using equations (2). 6 Bk's ara cubic B- spline functions. Tha first chapter ol part I deals with them at length as regards modeling trajectory formation.

8 CD O O Figure 13. Bicubic spline surface patch. Left: control polyhedron. Center spline curves on the surface. Right: shaded image of the patch.

Shape and Movement 307 The Inverse problem consists of computing the control points which induce a surface patch to fit a set of surface points. The problem is more complex but methods for solving It have been devised [Barsky and Qreenberg 1980]. 3.5. Local Surface Geometry The polyhedral representation of a surface by means of a trlangulation makes explicit the connectivity of the surface: which points are neighbors of any point. However it dees not provide other local features of the surface, such as the orientation of the normal and the Gaussian curvature which are very significant either from the motor or from the perceptual point of view. We can estimate this information by fitting locally the data with a smooth continuous surface of the form (1). Let us assume that r(t1,u1) Is the point around which we wish to get information on the local shape. The two coordinate curves that cross themselves at that point are (3) r(t,ui)andr(ti1u) and the corresponding tangent vectors v. and vu are the derivatives of the two curves computed there. From the two tangents we can compute the normal vector by means of the vector product (4) n - vt * vu The knowledge of the normal vector is important in many cases, for example when shading in computer graphics or when selecting an approaching path toward the surface of an object. However, in other cases this is not sufficient: for example, if we wish to "place" the hand on the surface of an object after having approached it, we need to have some information about the local shape of the surface. This requires a number of concepts of differential geometry that we are going to summarize here (the reader may refer to Hilbert and Conn Vossen [1952]). To begin with, consider the various directions in which one could proceed away from a point on a cylindrical surface. In the direction parallel to the cylinder axis the path9 is a straight line, in the perpendicular direction the path is circular, and in intermediate directions the paths are elliptical. If the surface path is saddle-shaped, however, some paths would arc In one direction while others

308 R. Boccardo et al. would arc in one direction while others would arc in the opposite direction. As the examples show, normal curvature depends on the type of the surface and on the orientation of the cutting plane. A fundamental properties of smooth surfaces is that the two cutting planes that determine the minimum and the maximum normal curvatures are mutually perpendicular (the two curvatures are called "principal curvatures")10. For example, the two principal directions of a cylinder are parallel and perpendicular to the axis of the cylinder, and the corresponding principal curvatures are zero and equal to the inverse of the cylinder radius, reepectively. The combination of the signs of the principal curvatures determines the local behavior of a surface: Dn a saddle surface, the principal curvatures have opposite signs, on a pebble or inside a cup they have the same sign11, on a developable surface (a surface similar to a smoothly folded and twisted sheet of paper) one of the curvatures is zero (in the particular case of a planar surface both curvatures are zero). Such a qualitative characteristic of the local shape Is captured by the sign of the Gaussian Curvature (defined as the product of the two principal curvatures): it is negative In the saddle case, it is positive in the convex/concave case, and it Is zero in the oylindrical/planar oase (see figure 14). The Gaussian curvature, as well as other local differential features can be estimated - in an array of empirical data - by means of the classic techniques of numerical analysis: fitting a Figure 14. Gaussian curvature of a surface pstch. In A and B it is zero, in C it is positive, and in D it is negative. 9 By 'path* we mean a ourve determined by the Intersection of the surface with a normal plane. The local curvature of the path Is the "normal ourvature" of the surface lor the particular cutting plane. 10 The perpendlculerlty of the two principal directions is related to the Euler's theorem which states thet cR • cos delta c(+ sin delta c2 where on is the curvature of a normal section, d and c2 are the prinoipal curvature* and delta Is the angle between the plane of the normal section and the plane of the first principal section. Note thet the normal curvature, as well as the principal curvatures, are signed quantltias. " This dcas not distinguish between concavity and convexity. In general, given the parametric equations of a surface, the "Inside" can be distinguished from the

"outside* only if we can rely on a particular choice of tha parameters, for example a choice according to which the normal computed in (4) is directed outward.

Shape and Movement 309 smooth model to the data (e.g. a bicubic polynomial patch) and then computing from it all the necessary partial derivatives. The local differential features have been used by several authors to characterize and segment a smooth surface in homogeneous surface patches: Nackman [1984] isolates non degenerate critical points of surfaces - local maxima, local minima, saddle points - and forms critical points configuration graphs, according to an adjacency criterion. Surface segments are extracted as slope districts, i.e. as regions bounded by graph cycles. A topographic primal sketch has been proposed by Haralick et al. [19831 and Watson et al. [1985] who assign topographic labels'2 to surface points on the basis of the Hessian matrix of a local polynomial approximation. A related approach is that of Besl and Jain [1985] who used the sign of the Gaussian curvature and of the mean curvature to define eight surface types 12. Asada and Brady [1984] proposed a surface primal sketch that combines information on significant surface discontinuities. Another method of object surface characterization is by means of Extended Gaussian Images - or Needle Maps [Horn 1984]. Needle maps are obtained by mapping onto the Gaussian sphere the unit normals of the object surface. Although this characterization may be useful for object recognition (specific patterns can be conveniently matched using the Hough transform), it is useless for classification and local shape analysis because it looses the notion of adjacency. 4. CONSTRUCTIVE SHAPE ASSEMBLY Programmed-to-bend-in-the-breeze computer-generated blades in a computer- animated field of grass may suggest how sophisticated the technology for generating shapes with the aid of the computer is already. Physicians, manipulating CRT images, can visually probe the brains of patients without having to perform exploratory surgery, thus experimenting what one would inevitably call computer assisted insight. Not only can one reproduce nature with a computer but also do something one could never do with a camera. It is not the intent of this section to focus the attention on computer graphics techniques for generating images. In the paragraphs to come we summarize the evolution of geometric modeling and related technologies, briefly describe the functional capabilities of a constructive solid modeling system, and present examples of constructive shape assembly as we perceive it for the purpose of this book. 12

The elgth surface types are: Peak (G: + ,M:-), Ridge (G:0,M:-), Saddle Ridge (G:-.M:-), Plane (G:0,M:0). Saddle valley (G:-,M: +), Valley (G:0.M: +). Cup (G: + ,M: +), Minimal Surface (G: + -,M:0). The symbols < - 0 + > refer to the sign of the Mean curvature (M) and of the Gaussian curvature (G).

 It seems pertinent here to underline the two different fields of action of computer graphics and what is called geometric modeling. To make it clearer we will use an example in industrial environment: let us consider the computer aids for the design and generation of desired objects shapes. The computer aided design (CAD) software provides a kernel of computer programs to implement computer graphics on the system plus application programs to facilitate the engineering functions of the user company. CAD involves any type of design activity which uses the computer to develop, analyze, or modify an engineering design - It increases the productivity of the designer, it Improves the quality of the design, the standardization of the drawings and the documentation of the design, it creates a data base for manufacturing. In computer aided design it is the geometric modeling system which is concerned with the computer compatible mathematical description of the geometry of an object. The mathematical description allows the image of the object to be displayed and manipulated on a graphics terminal through the CAD system commands. Using geometric modeling the designer generates basic geometric elements, scales, rotates and transforms them, and finally joins them into the desired shape of the object to appear on the interactive computer graphics system. It is here that computer graphics techniques loom on the horizon to put at disposal an user-oriented system to create, transform, and display data in the form of shapes or symbols. Computer graphics strikes directly at the problem of effective and efficient display of information In graphic form: it allows convenient communication through charts, diagrams, and pictures. However, all the issues regarding the knowledge of spatial and movement phenomena are not part of computer graphics. 4.1. Geometric Modeling Systems and Solid Modeling Geometry - the ancient branch of mathematics that deals with shapes and spatial relations - is central to the definition and design of objects. Early times saw artisans carry geometry In their minds or rely on physical objects' models. Mass manufacturing made then drawings become the primary medium of geometric specification. We all have a concept of geometry bound to our first years of study at school. Lines, triangles, circles, areas, lengths and so on have filled our geometric world. A world which has progressively grown and organized itself with a precise structure: two Cartesian axes - one perpendicular to the other - going to infinite like an Imaginary universal board (everybody is bound to remember the figures that went out of the black rectangular board invading parts of the wall too). A world which has an evident limitation: bi-dimenslonality. The concept of bi -dimensionality is well rooted in the representation. The drawing began to illustrate objects and animals thousands of years ago and only after the year one thousand the perspective - the first expression of tri - dimensionality - was introduced. Up to now, In industry, we very often refer to bi - dimensional views or sections of the object to be manufactured. The designer, though he has the 3D Idea of the real piece, is traditionally used to translating this idea in a series of drawings : front view, lateral view, perspective view, etc. The shop floor operator, reading the drawings, gets all the necessary information to carry on the correct sequence of

 operations on raw material for the desired product. Let us assume we want a groove on an aluminum block. Tha operator reads the drawings, reconstructs the 3D model In his mind, defines a working strategy and finally operetes with the cutter. The 2D drawing is in-between two 3D Ideas. Material constraints have forced people to use 2-D representations. The concept of a chair, for example, is linked to several other concepts such as that of having four legs and a back to lean against, of having a certain dimension, a certain consistency, of being used as a place where to sit, and so on. However, when you have to represent a chair on a piece of paper you necessarily filter the Idea and you give this set of notions a certain number of projections and sections. This seems to be the best way of communicating the Idea of the object to be represented to the others: the verbalized concept would not be appreciated neither by the designer nor by the shop floor operator. Hence the drawing has always been the important link between designers and shopfloor operators. The advent of computers, plotters and CRT displays led to a wave of computerization of drafting in the 1970's. Today, keyboards and cursors replace pencils and development sketches are ephemeral CRT images. The link between a drawing and 2D representations is loosing Importance in the industrial environment: the computer allows the designer to conceive pure 3D concepts, perhaps nearer and nearer to neuronal representations. Although we are still certainly far away from being able to directly transfer an idea from mind to machine, it is already possible to represent objects as unions and

Intersections of other objects (namely, primitive solids such as cones, spheres, cylinders, etc.) or as sweepings of a planar shape moving along a spatial trajectory13. Computer based systems for the generation of shapes need well - defined symbol structures (or representations) of the abstract solids which are designated to model spatial structures. There are several different methods of representing the objects in geometrio modeling. The basic form usas wire frames. The advanced method of geometric modeling is solid modeling In three dimensions. This latter uses solid geometrical shapes called primitives to construct the object. The term solid modeling indicates that body of theories, techniques and systems based on informational!)/ complete representations of solids so that any well-defined geometrical property of any represented object may be calculated automatically. The three most popular schemes of solid modeling are the boundary representation, the constructive solid modeling (CSQ) representation and the sweeping representation. (a) The boundary approach describes an object in terms of Its enclosing surface, the latter being composed of faces (i.e. surface patches). (b) The CSQ way of representing shapes Is through the specification of a binary tree or graph, each of the nodes being either a primitive solid, a movement or a constructive operator. (c) The idea of the sweeping representation is that solid sets of points can be represented as the Cartesian product of an area set and a trajectory set - the former moving along the 13 For example, a cylinder may be generated by sweeping a circle along a aegment perpendicular to tha baaa.

312 R. Boccardo et al. trajectory sweeps out the represented solid. A very general way of sweeping Is implemented by the generalized cylinders, the trajectory can be curved and the moving section can change its shape during the movement. In particular, generalized cylinders are the basic primitives of the model-driven vision system ACRONYM [Nevatla and Binford 1977, Brooks 1983). An even more general way of sweeping is obtained when the moving shape is solid, not planar. This is the case, for example, of the 3-D version [Nackman 1984] of the Medial Axis Transform [Blum 1987] which uses spheres moving on a skeleton. This kind of paradigm, In particular, was used by O'Rourke and Badler [1979] for representing the human figure - the Bubble Man. In the following section, however, we focus our attention on CSG. 4.2. Constructive Solid Geometry (CSG) The basic idea of the CSG representation Is that complicated shapes and solids can be obtained and represented as the result of a simulated assembly, i.e. a process where (i) primitive shapes are

instanced, (il) primitive shapes and sub - assemblies are moved in the assembly space, and then (Hi) are put together. This requires primitive shape operators, movement operators, and regularized Boolean operators (Requicha 1980). The CSG representation of a solid object Is an ordered binary tree whose non terminal nodes are the operators (either rigid motion operators or regularized set operators) and whose terminal nodes are either descriptors of primitive solids or descriptors of rigid motions. Let us see the BNF for the tree in question: <CSG tree> ::« <primitive solid>| <CSG tree> <set operator name>| <CSG tree> < motion node> < motion descriptors > CSG is perhaps the best understood and currently most relevant representation scheme for solids14. 4.3. An example of CSG: PADL-2 PADL-2 (Part & Assembly Description Language - version 2) Is a collection of software modules for use in industrial and research environment dealing with problems regarding solid modeling systems. 14 Extended CAD models have been delined lor object recognition. The basic volume-surface-edge-vertex model is enriched by the addition ol redundant pointera and other data structuree. 30PO, tor exampia, is ona ol such systems IBolles et al. 1983, Horaud and Bolles 1984] which extends to 3 - D the 2 - D local feature locus ideas discussed in Bolles and Cain |1982|.

Shape and Movement 313 The package has been developed in the Production Automation Project of the Department of Electrical Engineering at the University of Rochester [Brown 1982]. It is intended to provide representational and computational facilities to different geometric modeling systems with an eye to Integration in the manufacturing environment. The PADL-2 primitive solids are: Block (parallelepiped), Sphere, Cylinder, Cone, Wedge, Torus. Primitive blocks can be Instanced in many copies. For example small cone « CON(r -1 ,h « 2); big_cone - CON(r«10,h « 20); long_cone « CON(r«1 ,h « 30); are three instances of the same primitive "cone" with different parameters. Primitive blocks have an intrinsic system of reference: e.g. the cone origin is at the center of the base and the z - axis is directed toward the vertex. Primitive solids can be moved by means of translations or rotations with reference to anyone of the three Cartesian axes. For example translated_cone-MOVE small_cone BY (MOVZ-2); turned_cone « MOVE translated_cone BY (ROTX-3.1416); turns the small_cone upside down15. Another way of dealing with movements of primitive blocks or partial assemblies is to use multiple Cartesian systems of reference and to "attach" solids to them. For example translated CS - XFORM LAB BY (MOVZ - 2); rotated_C5»XFORM translated_CS BY (ROTX-3.1416); turned_cone « small cone AT rotated CS; generates the same cone as before in a different way: the LAB coordinate system is translated and rotated, and a small cone is instanced and then attached to it. Partial assemblies can be built by applying (regularized) set operators to primitive solids and/or partial sub-assemblies. These operators are: UN(ion), INT(ersection), DIFJference), ASS(embly). (assembly is a special type of union). For example, the sequence 15 The translation parametera are in arbitrary units and the rotation parameters are in radians.

314 Ft. Boccardo et al. bottom ball-MOVE SPH(r«l) BY (MOVZ-2); top_baI-MOVE bottom_ball BY (MOVZ-1); ice_cream-turned_cone UN bottom_ball UN top_ball; generates an ice cream shaped object. Assemblies or sub - aseembllea can be parametrized, by means of the GENERIC construct, that allows the user to define (new) object templates: GENERIC ext name(ice_cream); cone-MOV? CON(r»size,h«3*size) BY (MOVZ-3'size); u_cone-MOVE cone BY (ROTX-3.1416); b_ball-MOVE SPH(r-slze) BY (MOVZ-3*slze); t_ball«MOVE b_ball BY (MOVZ-size); ice_cream « u_cone UN b ball UN t_ball; END; Generic objects, as well as primitive objects, can be Instanced in many copies, with different dimensions. PAM: the PAdl Man A specific application of solid modeling, studied and developed at DIST, is the definition of the model of a humanold using PADL - 2 (PAM - PADL MAN). Representing man has always gained in the eye of the artist a considerable significance. In a sense, the application of solid modeling concepts to the human body Is a natural idea tor an artist. In particular, we took Into account the ideas of Oskar Schlemmer, who taught a 'course on Man* at the Bauhaus [1971]. Oskar Schlemmer saw man as 'that psycophysical whole to which his chief obligation as an artist lay'. In his syllabus 'Subject of instruction: man' he tried to suggest the totality of the concept 'MAN* developing his formal, biological, philosophical parts, corresponding to the graphic representation, scientific structure and transcendental world of Ideas. Figure 15 gives an example of the 'solid modeling" approach whloh was implicit In Schlemmer's conception. The PAM model is a preliminary attempt to put Schlemmer's ideas Into computational form. PAM uses two types of generic blocks: the truncated cone and the sphere. From these basic blocks solid sub - assemblies are generated that identify the different parts of the body (limbs, head, torso, etc.).

A sphere Is located in all the articulated joints. Figure 16 shows PAM In the standard posture. PAM is an articulated solid because the joints of the structure are identified by Cartesian systems of reference and can be rotated In any way. Therefore, the Interface between PAM and a computer animation language, like NEM (see a preceding chapter), consists simply of interpreting as PADL rotation commands the homogeneous matrices managed by NEM. The use of apheres at the joints is convenient because it guarantees a smooth shape at the joint/linkage Interface for all the physiological movements.

Figure 15. A Schlemmer's Man. \ I i I Figure 16. PAM: the PAdI Man.

This Page Intentionally Left Blank

PART IV TIME For purposive actions, time is not an independent variable which flows continuously. Purposive actions always imply a multiplicity of interacting actors and time is a stream of messages, events, decisions which constrain them. In other words, time is concurrency, co-operation and/or conflict. The spatial and geometric aspects of concurrency are examined in the previous parts of the book. This part focuses on the timing aspects, reviewing some general concepts of representation - in the first chapter - and investigating a particular type of timing structures - musical processes - In the second chapter.

This Page Intentionally Left Blank

PART IV Chapter 1 CONCURRENT PARALLEL PROCESSES Giuseppe Marino, Pietro Morasso, Renato Zaccaria CONTENTS

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasco (Editors) © Elsevier Science Publishers B.V. (North-Holland), 1986 321
CONCURRENT PARALLEL PROCESSES Giuseppe Marino, Pietro Morasso, Renato Zaccaria The nature of time in skilled, purposive behavior is determined by concurrency and parallelism. Time is not an independent

variable and the chapter diecusees how its structure Is linked to message - passing. Attempts to formalize concurrency and parallelism are distributed in the history of Computer Science. This chapter reviews the evolution of some concepts - processes, semaphores, monitors etc. - with particular regard to their motivations and limitations. 1. THE NOTION OF CONCURRENCY IN COMPUTER SCIENCE Concurrency is one of the major building blocks of human purposive behavior and also one of the major sources of its complexity. However, it is a subtle notion that still lacks a satisfactory formalization even in man made machines like digital computers. In this chapter we address some aspects of concurrency as they have been maturing in Computer Science. The approach is quite informal and covers just a fraction of problems, methods, and techniques. In a sense, this chapter is at the same time a discussion section for all the previous chapters and an introduction to the following chapter. Computer Science (CS) is the science that deals with systems for the automatic solution of problems. How to formalize problems, and how to reason about them is also part of the discipline. Therefore, CS is a wide science: it includes theoretical topics (theory of computation, formal languages etc.), methodological topics (architectural design, software engineering, operating systems etc.), technological topics and so on. As any Computer Scientist knows, most of these topics lead apparently Independent lives. Computer science involves a great amount of human creativity as regards both using and designing computers, and we are mostly interested in which kind of languages and models are used by man when dealing with a heterogeneous, sparse, highly complex1, typically concurrent and asynchronous system like a computer or a system of computers. ' Computer systems are among the few man made systems whose complexity cennot be understood by a single man at a reasonably line lew! of detail.

 In the following we attempt to single out some subtle oonnections between the basic concepts of time and parallelism in CS and purposive behavior of a robot. The strongest connection between CS and Anthropomorphic Robotics may be found in the history of the (linguistic, semantic) formalizations of computation and computing systems. Until fifteen years ago, computation was intended mainly in the numerical sense. Afterwards, particularly as a significant contribution of Artificial Intelligence, the same word also acquired a symbolic meaning, such as theorem proving, planning, knowledge representation, and so on. Regardless the specific computation, a Computer is a particular Robot with no sensor nor motor activity, whose behavior has been (and is) described in terms of temporization, states, concurrency, interactions, synchronizations etc. of more or less "hard" entities: gates, processes, actors, processors, messages, data and so on. Another strong analogy between men and computers can be seen in Real Time computer systems. Like living beings, they are organized collections of computational entities (processes, devices, etc.) which cooperate by exchanging information and are able to react in a purposive way to the demands of external asynchronous stimuli. Moreover, real-time systems react somehow even when facing conditions not explicitly considered in the design phase. Such behavior may be clever or dumb, useful or wasteful, but emphasizes the fundamental idea that a complex system is able to produce answers, although not necessarily correct, also to questions that were never defined before. The "non foreseen" answer is produced by the interaction among the sub-systems, every one of which is known, but whose full set of possible Interactions is not known a priori. As any user of a big operating system knows, a lot of unpredictable things can happen in non standard circumstances: though any process and any interface is (almost completely) known, the set of all possible interactions Is beyond a possibility of clear understanding. This is considered a drawback for an operating system - one of the goals of a good operating system is indeed to restrict as much as possible the impredictability of its behavior. For a skilled, cognitive robot, on the contrary, such a behavior is necessary and powerful. A robot, in fact, unlike an operating system, is an open system, that has to cope by definition with a continuously changing environment and, possibly, with the changes of its own knowledge base. Although the name - Computer Science - is recent, CS comes from far and we can single out three main historical components: 1. An engineering activity, whose aim is the design of computing machines. 2. A mathematical activity, which studies the properties of computational problems. 3. A cybernetic activity, which alms to capture intelligence and encapsulate it into machines. The engineering component is probably the oldest one: The first computing machine is said to have been made by Whllelm Schickard in 1823, the same year In which Blaise Pascal was born. The other steps on the road towards the electronic digital computer are well known: The computing machines by Pascal and Leibnitz, the Analytic Engine by Charles Babbage (the first true programmable machine - near 1840), a similar device by his contemporary Qeorg Scheutz, the first electric computer (based on binary representation of data and programs) by Konrad Zuse in the 40's.

Concurrent Parallel Processas 323 It is worth noting that during such a pre - history the computing machines were conceived as strictly sequential devices: a sort of emulation of the way in which men carry on (or think they do) simple computational problems. This concept is clearly detectable in Babbage's first description of (what we call now) algorithm, that he defined as a sequence of operations. This definition has crossed one century and still holds. The mathematical ancestor of CS had an independent evolution until the 30's. The mathematical definition of algorithm grew on a more abstract field than numerical computation, in particular on the problem of devising formal mathematical systems, based on logic and predicate calculus, able to automatically solve any question about a particular system. This is, of course, a field very related to theorem proving and planning - a section of Artificial Intelligence (Al). After the 30's, robust formulations of computational problems were developed and formal definitions of algorithm and of computing machine were investigated. The most important philosophical result was an improvement of the consciousness about the nature of machines and algorithms - what they can and what they cannot do2. This theory, however, emphasizes the sequential, synchronous aspects of algorithms - think for example of the well known Turing Machine. Aleo the formal description of the computing system, for example by means of Predicate Calculus, Is a collection of statements with no implicit notion of time nor time sequence. The third ancestor of CS - cybernetics - evolved towards computational problems in the 50's when it partly relinquished its original approach, based on feedback theories. Starting in the early 80's, Al inherited the aims of cybernetics - a new form of cybernetics focused on computational models. Al is a composite field, partly based on the previously mentioned formal theories of computation, and partly on heuristics. It is in this field that a different approach to concurrency is growing. The concept of Actor, introduced by Carl Hewitt, involves a different point of view on concurrency. In an Actor System concurrency is a way for representing information and knowledge, whereas in "traditional" concurrent systems (e.g., a multiprocessor), concurrency is a physical constraint to be managed in the best possible way. 2. THE NOTION OF TIME IN CONCURRENT SYSTEMS Since the time of Aristotle, the notion of time has been associated with the idea of an arrow which flows uniformly from the past to the future - the independent variable upon which the whole structure of reality is built. From this point of view, time imposes an absolute ordering among all the phenomena at any level of scale. A universal master clock is the epistemological device which implements this idea and gives an unambiguous meaning to timing relations: before, after, simultaneously. 2 Mostly influential Is tht incomptttontss Iheoram by Kurt Gooel [1931] which states, among other (hinge, that no algorithm can exlet whose Input is any statement about Integers and whose output tells whether the statement Is true. This wes the loundation of the theory of computability, to which many authors contributed - Turing, Kleene, and Church among others - not only by llnding out many other non-computable problems but also Investigating fundamental definitions of machine and algorithm.

 In physics, we know that this kind of assumption is constrained by the speed of light: The universal description of physical phenomena is acceptable if the relative speeds at play are much smaller than the speed of light. However, the physics involved in the human experience of every day is far from the conditions in which the relativistic Einsteinian effects become significant and indeed the old Newtonian mechanics is the universal tool for describing our mechanical experience. Nontheiess, it is not the speed of light per se which destroys the universality of the Newtonian time, but it is the role of light as the medium with which different observers can exchange messages. Light, however, is not the only possible medium. In other systems we may have different media. For example, in the brain the messages travel on the neurons at an insignificant fraction of the speed of light and in computer networks messages are transmitted according to slow transmission protocols. In these cases - and also in many others which are part of our daily experience - the time scale varies widely but it is easy to understand that what matters are not absolute values (in meters per second, bits per second, and so on) but ratios. If we focus on the distinction between communication and (local) operation we recognize that the master clock paradigm - synchronous operation - is feasible only when the time spent communicating is much smaller than the time spent operating/computing. When the message-passing activity becomes a significant part of the business, the universal time breaks down and each individual unit tends to acquire an individual time structure, its individual clock: asynchronous operation. Even if literally possible, large systems, e.g. computer networks, cannot afford to be synchronous. Synchronicity would over - constrain them, stop their evolution.

A synchronous system is closed, it cannot communicate with anything else except itself. In a closed system the order imposed by time is an illusion, there is no reason why the order of events cannot be reversed. The puzzling and illusory nature of (Newtonian) time was emphasized indeed by many, from Aristotle to Einstein and Bergson. On the contrary, asynchronous systems are open. We discussed already, in a previous chapter, the implications of open systems from the point of view of knowledge, reviewing the ideas of Hewitt. Here we wish to stress the implications with respect to the notion of time. In an open system, time is not an illusion, it is an irreversible partial ordering of events which is determined by the nature of message-passing among the actors. We also wish to emphasize the similarity between this notion of time in open systems and the notion of causality that is defined in qualitative reasoning and is reviewed in the chapter on planning. In both cases, indeed, the driving concept is the asynchronous message - passing activity of a large population of actors. If we look - from the outside - at the performance of a large population of interacting agents, the global time does not appear as an independent variable but as a structure which is built (or destroyed) by the mutual interactions. Think, for example, of the musical time: in five mechanically measured minutes of a Bach fugue we may single out a whole world of timing structures and meta - structures which are meaningless in a Newtonian framework.

Concurrent Parallel Processes 325 Finally, we wish to touch upon the closeness of the notion of time (as the side-effect of message - passing in open systems) to the notion of time which is formulated in the context of the new thermodynamic theories, focused on the behavior of systems far from the equilibrium state [Prigogine 1973]. The irregularities and instabilities of systems of this kind, such as the atmospheric weather, the cerebral electrical activity or the vorticous motion of fluids, are far from random3: They exhibit a lot of structure which tends to isolate individual subsystems and make them relatively autonomous. The Benard instability is a good example. This happens to a liquid which is warmed up from below: At a certain point, convection streams are created which result from the interaction between heat flow and gravity. Each convection cell has an enormous number of molecules, of the order of 102'. The non equilibrium creates coherence, aggregates subsystems, allowing macro interactions - messages - which would be meaningless in the equilibrium state. However, we may wonder whether or not (intelligent) skilled behavior can be reduced to the pattern formation effects of inanimate matter - like vortex formation. The difference, in our opinion, cannot be clear cut because, after all, a pool of neurons and muscles is a complex aggregate of material elements. Nevertheless, we think that a qualitative difference is the typical intelligent characteristics of pushing levels of meta-structure in an indefinite (open) way. The lower levels are determined by the physical nature of the neuron pools whereas the higher levels represent more and more abstract goal - oriented structures. Such an infinite recursive nature of skill motivates, among other things, the computational analysis of human movements developed in this book. The different computational modules that have been discussed do not necessarily correspond to rigid, anatomic localizations, but to coherent operational aggregates whose meaning is in the pandemonium of message - creation and message - passing. 3. THE EVOLUTION OF CONCURRENCY Although the three ancestors of CS mentioned in a previous section show a consistent tendency to avoid concurrent formulations of problems, each one has contributed to the issue of concurrency in a peculiar way. As regards the engineering aspect, two main problems had to be addressed: i) writing reliable operating systems, ii) managing inherently parallel architectures, like multiprocessors, networks or architectures based on specialized parallel sub - systems, like array processors and so on. Both cases were faced by introducing the concept of Process, as a distinct algorithm acting In parallel, interacting with other processes and with the environment by means of a set of semantically defined primitives. It is worth noting that no clear cut, formal distinction can be 3 Complex non linear systems, operating far from equilibrium, seem to evolve under the inlluenoe ol strange abactors - e.g. seasons in the case ot atmospheric weather. Strange attractors datermine the behavior ot a system - its future - In a non deterministic way, that tends to fill up the space of possibilities in an infinitely dense way, according to a fractal geometry.

 made between the "process" concept and the "actor" concept. The difference is in grade and context. In an operating system, concurrency - I.e. the interactions among processes - is "controlled" or "limited* by the language, which is asked to guarantee a minimum safe number of interactions (and in many cases, to discourage others). Processes are usually in small number and are statically determined. On the contrary, the philosophy of an actor system Is to "encourage" not only the proliferation of actors, but also their exchanges of information assuming that solutions may emerge from interactions not defined a priori. The "mathematic" and "cybernetic" ancestors of CS gave a strong contribution to the formal aspects of programming languages [Treleaven et al. 1982). The most interesting one, from the concurrency point of view, is the one related to the theory of Applicative or Functional Languages*, which is a powerful and clean formalism for defining algorithms and machines. This field has discovered the amount of implicit parallelism inside algorithms, giving methodologies for designing highly parallelized machines able to automatically exploit the amount of parallelism of a specified algorithm written in functional style. Machines of this type should be able, for example, to execute concurrently the multiplication of a matrix with a vector, or the derivation of inferences in a production system. No such machine is available yet, but this is the goal for the next- generatlon computers. From the point of view of modeling purposive action, Implicit concurrency Is strongly related to all computation - intensive problems - both numeric and symbolic - like vision, kinematics, dynamics, inference making and so on. Many proposed models show appealing features like redundancy, total asynchrony, independence from the actual number of processors involved, tunable parameters of execution strategies [Qaglio et al. 1985]. Hardware Evolution One of the reasons of the absence of concurrency in the early development of CS is the big difference In the scale factor between the objects managed by hardware and by software. The basic hardware unit - the gate - is obviously apt to work in parallel. Groups of gates - digital circuits - can be synchronized either using master clocks or well defined protocols, like "hand-shaking". On the other hand, software deals with jobs, within which it Is rather difficult to detect conourrent activities. Moreover, in the eariy oomputers I/O units were so slow and limited that anything more complex than "a job at a time" was useless. The forces that pushed technical evolution during the 60's, the "hardware challenge" period, were due to the need to adapt a fast growing hardware technology to a software philosophy standing still in the "one-thing-at-a-time" habit. Since the formalization of sequential activity is much simpler than that of parallel ones, It makes sense that computing machines tend to grow in the line of sequential paradigms. The example of microprogramming is typical: Microprogramming can be regarded as a tool for

4 Applicative, or Functional Languages ('Pure* Lisp. FP, Sasl and othars) are languages based exclusively upon the mathematical concept of recursive functions. Algorithms are pure lunctlonal applications.

Concurrent Parallel Processes 327 applying software (sequential) techniques to hardware "programming", leaving out all the complexity - and the concurrency - of basic hardware components. The "Software Crisis" of the early 70s is indeed a consequence of the challenge made possible by a faster and more powerful hardware. We can identify two related aspects: On one side the unsuccessful experiences in the management of very large software projects pinned out the high non - linearity of scale factors in software projects; On the other side, the difficulties in implementing operating systems powerful enough for that hardware unveiled the lack of sound theoretical bases of concurrency. The two aspects are related: An operating system is also a very large program, and its project and implementation requires the coordinated effort of a large number of people, with the same problem of Management and Engineering of all big software projects. Theoretical studies about the basic concepts of concurrency were driven by that crisis, and it is not by chance that some of the researchers that contributed to understand and formalize the issues (E.Dijkstra, C.A.R. Hoare, and P.B.Hansen among others) had practical experiences of operating system design at the beginning of their professional curricula. Several basic concepts such as synchronization among concurrent activities, risks of deadlock when "processes" compete for resources, the ubiquitous communication principles, and so on, could get their initial formalization, although the underlying computing structure is still conceived as a sequential machine. Software Evolution On the side of pure sequential programming, the "Software Crisis" stimulated the outgrowth of new high level languages. (It was observed that the number of lines of code per programmer per day is roughly a constant.) The project of high level languages, with higher and higher levels of abstraction, triggered in turn an activity of formal definition and understanding. We can single out a series of oycles, where innovative steps are followed by systematic steps - either standardization or theoretic formalization. Let us list a few example: Definition of Algol 60 - an innovative step that produced a powerful language, albeit plagued by problems due to the semantic ambiguity of its definition. Proposals about formal semantics. Implementation of the first complex operating systems (OS 360). Problems with deadlocks and synchronization. Critical contributions by Dijkstra, Hoare, Brinch Hansen and others. Formal definition of basic principles of concurrent systems. Definition by Wirth of the Pascal language. A simple language, devised for teaching, but easily adapted to many professional environments. Axiomatic definition of the semantics of Pascal: not only the basis for unambiguous definition and standardization, but also a verification of the soundness of the language.

328 G. Marino, P. Morasso and R. Zaccaria Definition of the Ada language, according to the requirements of the US Department of Defense. Ada, which is deeply influenced by Pascal, was developed together with its formal semantic definition from the very beginning. One of its aims was to support both sequential and parallel programming in a standard and safe way. The Ada language dees not follow the paradigm of cyclic evolution: it is both an innovative step and a standardization which tends to "freeze" its own innovations: This is considered by many [Parnas 1984] as the major flaw of the language, but the story is still in progress. Anyway, Ada can be considered as one of the milestones which mark the convergence of the autonomous evolution of sequential programming and parallel programming. About Notation When people have to deal with complex problems - problems that exceed the capacity of their short term memory - they tend to resort to notation. And notation, indeed, is very important when attempting to understand the behavior of concurrent system. According to Brinch Hansen [1977], the main principles of programming languages (applied to concurrent systems) start with some basic Ideas, subsequently refined and defined with precision. Then we need a notation, possibly a terse and simple one: A good notation can generate a formal language, and in the end It may become a programming language, provided that an efficient and safe implementation is achievable. Several general concepts of concurrent programming were developed: semaphor, monitor, message, rendez-vous, and so on. Almost all basic tools have been defined axiomatically extending the framework of some sequential language. The basic idea in concurrent programming is the definition of how processes communicate, or how processes can share common data. A suitable mechanism must be reliable, safe and easy to use. The first formal notation is due to Dijkstra [1968] - the famous concept of semaphores a signal to control access to a shared variable in a one-at-time way. The communication component of concurrency was hence identified with data sharing between processes. The monitor concept is based on the same idea, trying to express a higher level of abstraction in the representation of the shared datum. It can be seen as a guardian, a piece of code or, in software jargon, an abstract data type, that specifies the ways of accessing and changing given structured data. Any entity - a process - that needs to interact with these data must seri- allze its calls

(apparently similar to procedure or function calls) to the monitor code. The monitor concept and its evolution may be taken as a paradigm of the principles stated above. One of the first problems in concurrent programming was the management of shared variables, i.e. variables used and modified by more than one thread of computation (a "process"). Besides the simple and effective mechanism of "semaphore" which offers a tool for synchronization only, the concept of critical region allowed to track down the (possibly concurrent) usage of shared variables. It was proposed to put aside in an independent module all references

 to a given shared variable in order to circumvent some of the problems of critical regions. A clean implementation of monitors in terms of one or more semaphore was also devised, either for the mono - processor system (all processes implicitly serialized by an hardware mechanism, such as the "interrupt queue") or for the multiprocessor scheme (some autonomous CPU's accessing a shared memory). So far, both monitors and critical regions are simply abstractions of basic techniques applied for solving concurrency problems by any programmer in whatever language; some formal rules for proper synchronization can be stated either informally or using an axiomatic formulation, i.e. we go from basic ideas to formal definitions. Brinch Hansen [1975], devising the Concurrent Pascal language, introduced the monitor as a semantic unit of its own, and gave it an (almost simple) syntactic form. Now we have a notation, something that helps a system programmer to work at a higher level of abstraction. Given the exact definition of the monitor concept, a clean and secure implementation could follow immediately, and it was easily integrated in any compiler (and run-time support) of Concurrent Pascal. We can object, however, whether such an implementation is also an efficient one, particularly for multi - processor systems: The underlying synchronization tools are indeed over - constraining. In fact, radically different approaches are being investigated for the efficient management of concurrency in distributed, multi-CPU systems. Cooperation as exchange of messages was investigated later by Hoare [1978]. In this scenario, processes are disjoint entitles which send and receive messages whose semantic is defined in great detail. The "message passing" approach seems to find more success than the "shared data" one, although the full theoretical equivalence has been demonstrated. The ADA language, for example, uses a particularly powerful form of message passing, called rendez- vous, logically descending from Hoare's model. Strange to say, one might argue that the model of concurrency based only either on the "shared data" or "message" concepts is very restrictive: in fact, a general concurrent system should also consider the problem of dynamic creation and allocation of processes, self— modification of processes and so on, and not only the communication problem. As said before, this comes from the "operating system" point of view, which tends to enforce a safe and deterministic behavior. On a higher level than the concepts of semaphor and monitor, we can put the Actor System (AS) model. AS was introduced by Carl Hewitt several years ago [Hewitt et al 1973]. As we said before, no fundamental distinction can be made between "processes" and "actors". The main difference lies on limitations and on the general framework. First of all, the AS concept is a metaphor of computation as cooperation of entities, intended as a general framework for knowledge representation and problem solving in a decentralized, concurrent, asynchronous computational model. Therefore, in the actor model it is assumed that actors can generate other actors and can modify themselves. Messages have been chosen as a natural way of interaction. A message has a double meaning: an exchange of data, and an event which triggers the recipient actor. Events play the role of a distributed clock, hence no synchronization is required: Any

 actor is a self-contained entity with its own speed. Actors can be processors, processes, non homogeneous machines or devices and so on. AS is a fascinating model that bears an obvious similarity with various human activities, in general, and with purposive actions, in particular. The "open" nature of AS determines a "computational freedom" that is far from the framework in which processes, semaphore, monitors etc. have been developed. The counterpart of "freedom" is a lack of formalization in notation - albeit a deep work on the semantics of message passing is being carried out [dinger 1983]. Several languages and systems have been inspired by AS, in particular the NEM language that is described in a previous chapter. Another formalism which addresses concurrency and parallelism at a comparable level of generality is the Petri Net concept [Petri 1962], that is investigated In the next chapter as regards a special type of concurrency: musical processes. The Next Challenge No truly distributed system exists, at the moment. In this decade, the basic tools for managing concurrency have been moved into the field of practical applications, and many of the largest "mainframes" (or even "super-mini") are multiprocessors, usually consisting of 2 up to 16 CPU's and with at least a global ("shared") memory store. Concurrency, therefore, is buried in the interior of the operating system - the end user is not aware of it. User programs are written in standard sequential languages, only the apparent computing power5 of the machine appears different - increasing over the years. Sixteen - the practical maximum number of parallel CPU's - is a sort of "magic number" that reflects the intrinsic limitations of the shared resources approach which cannot be pushed to further parallelism. At the same time, hardware evolution is preparing yet another challenge, that is being played completely in the scenario of concurrency: the highly parallel systems made feasible by VLSI. Hardware technology is being pushed to such a point that we feel no practical limitations in what actually can be built6. From this challenging power different approaches to innovative architectures are emerging. Some approaches, while mastering hardware technology, attempt to fill the semantic gap [Myers 1982] between hardware and software, although in the framework of "traditional" architectures: The LISP machine Is a paradigmatic example. Others try to exploit a direct mapping from well understood algorithms and/or data structures to regular, repetitive VLSI structures. In this way it is possible to obtain useful solutions to specific problems - image processing, for example - which cannot be generalized. Systolic arrays are typical examples. 5 The computing power Is usually measured in MIPS: mega-instructions per second. s It has bean shown thai the physical limits ol VLSI are due to electromigration in metallic conductors and molecular level doping, but they are still well beyond ourrent techniques.

Concurrent Parallel Processes 331 Front runners try to devise generic architectures naturally suited for a wide variety of applications. To this end, however, innovative hardware per se is only part of the job: A key role, once again, is played by software. In particular, it is becoming more and more evident that, when stuffing a great number of Processing Elements (PE's) in a global architecture, communication tends to be more important than local speed and power. Attempts to enforce absolute global principles in highly distributed systems - global storage, global time, etc. - are bound to fail. Among the many directions that are being pursued, let us mention the following ones: architectures based on the message passing philosophy [Hewitt 1980], data-flow machines, reduction machines, and machines that, in one way or another, mimic neural networks. A data-flow machine [Dennis 1960] substitutes the concept of absolute time and global storage with the concept of data (packets) moving from PE to PE (without needing a global storage) and firing execution of code that transforms the data (a form of relative timing). Reduction machines [Berkling 1971] can be regarded as a way of mapping abstract objects (functions, lists, atoms, etc.) and higher level constructs {functors, functions that manipulate functions) over the basic data - flow structure. Finally, the motivation of building hardware structures that somehow resemble neural networks is the idea that intelligence is an emergent property of a complex system, like a neural net. This idea can be traced back at least to the models of perceptrons [Minsky 1969] and among the machines inspired by it we may mention the Wisard Machine [Aleksander 1983] - an all purpose pattern recognition device - and the Connection Machine [Wilis 1965]. The architecture of the latter, in particular, provides a large number of PE's connected by a programmable communication network. PE groups are connected together into data - dependent patterns - active data structures - that both represent and process the data. A conventional host computer directs from the outside this activity: While in a conventional machine the processor stores data in a passive memory, the host computer of a connection machine stores the data in an active memory that transforms the data in a distributed way. For machines of this kind the Newtonian notion of time is obviously meaningless, except for an external observer who watches the evolution of the computational patterns. It is also worth noting that these machines - as the human beings - cannot work in real-time if by this we mean a regular and reliable production of results: Fluctuations in the Reaction Times and Performance Times are intrinsic properties of complex open systems. In well designed systems as the human beings, however, the global behavior will be attracted by an autonomous rhythms that are essential for an efficient interface with the world.

This Page Intentionally Left Blank

PART IV Chapter 2 DESCRIBING AND PERFORMING MUSICAL PROCESSES A. Camurri, G. Haus, R. Zaccaria CONTENTS

A fugue by J.S. Bach Appendix B: Petri nets.

This Page Intentionally Left Blank

HUMAN MOVEMENT UNDERSTANDING P. Morasso and V. Tagliasco (Editors) © Elsevier Scisnoe Publishers B.V. (North-Holland), 1986 335 DESCRIBING AND PERFORMING MUSICAL PROCESSES A.Camurri, G.Haus, R.Zaccaria In this chapter we show how Petri Nets can be used to describe and perform musical processes. We have developed a system, called MAP, which, starting from a higher - level description by means of Petri Nets, produces sound sample files; the intermediate levels of representation are obtained via the CMUSIC package. Our approach makes it possible to easily describe hierarchy, concurrency and causality within musical processes and to perform them via DAC's. 1. INTRODUCTION One of the main r,oals of computer music research is the definition of languages suitable for the description of musical entities. Such languages require several features: affinity with the composer's way of thinking, different levels of representation of musical information, morphism mechanism between levels of representation, concurrent processing structures, operators to process and transform musical entities, high flexibility and so on. The basis for our research is the metaphor of musical object. We can assert that a musical process may be described as an evolution of one or more musical objects; the perceptual identification of such objects Is by no means either simple and unique: two distinct hearers (or the same hearer in distinct conditions) may have different perceptions of the same performance. We are therefore faced with the question of which and how many musical objects there are within a given musical piece. The answer is difficult and not univocal. One could describe a musical process as a (low level) sequence of sound samples: if one wants to avoid assuming a particular structure for musical objects, this is the only possible description of sound processes. Assuming structure means considering subsequences of sound samples at the lowest level of description as related in some way, and giving them unity and identity as musical objects. Therefore, music could be defined as the evolution of (possibly distinct) musical interacting objects: in the process of interaction, processes appear or disappear, modify or repeat themselves. The definition of objects as well as their transformations is a consequence of process of abstraction; the particular abstraction depends on the musical standpoint from which we observe musical processes. We can thereby use hierarchy, concurrency and causality as conceptual tools

336 A. Camurri, G. Haus and R. Zaccaria to understand the nature of objects interaction. Our proposal Is the use of Petri Nets [Petri 1976] to describe the time flow of one or more musical objects and their interactions. We have developed a system, written In the C programming language under Unix Operating System, called MAP (Musical Actors by Petri Nets), whose input is a linguistic Petri Net description, with sound samples files as output. MAP has been developed as a joint project between the Department of Communication, Computer and System Sciences (University of Genoa) and the Agency for the Advancement of Technical Education (EMIT, Milan). MAP is an extension of the CMUSIC package [Moore 1962] developed at University of California, San Diego. The following sections introduce the basic concepts of music description by Petri Nets and briefly discuss the system. 2. MUSICAL PRIMITIVES AND OPERATORS USING PETRI NETS In his introduction to the General Net Theory (QNT), C.A. Petri shows how systems (in the most general sense of the word) and morphisms between levels of abstraction can be described by means of the formalism which is presently known as Petri Nets (PN's). We take the metaphor of musical objects and Petri's QNT to show that PN's are suitable for musical description, both for composition and performance (see Appendix B for basic PN terminology). In a PN, we can connect the concept of place either with observable musical objects or control objects, i.e. objects which do not have a direct musical meaning. On the other hand, we connect the concept of transition with transformation rules or physical processing; by "transformation rules" we mean musical operators as introduced in [Bertonl et al. 197B]. 2.1. Musical Primitives We have singled out five basic place-transition nets with which we can describe musical texts (see fig. 1): (a) SEQUENCE, to describe a sequential flow of objects; (b) ALTERNATIVE, to describe musical cases Involving non deterministic choices; (c) CONJUNCTION, to describe a linking or matching (In a logical sense) of different objects; (d) FUSION, to describe the joining of two or more objects giving one object, and, finally, (e) SPLITTING, to describe the generation of two or more objects from a single object. Let us introduce now four higher level operators to synthesize/analyze musical texts (see Flg.2). The first operator, which we call LINKER, makes possible the concatenation of two musical objects: the new place P derives from both P1 and P3 (see

Flg.2a). The second operator,

O^+X) a - SEQUENCE. b - ALTERNATIVE. c - CONJUNCTION. o V a-fusion Figure 1. Musical primitives. 0*>0 0HX>0 0 O 0MK3 OK3 0 © ©*»Gm>0 ©>KD 6 - UNLINKER ok) o^o O-h-O © © 0-H-K3 OyO 0 Cr^O Ok) c - SYNCHRONIZER 0 - DE SYNCHRONIZER Figure 2. Musical operators.

338 A. Camurri, G. Haus and R. Zaccaria which we call UNLINKER, has the opposite function (Fig.2b). The third operator, called SYNCHRONIZER (Fig.2c), puts together two concurrent musical processes synchronizing them by means of transitions; the last operator, called DE - SYNCHRONIZER (Fig.2d), separates two concurrent musical processes (from the temporal point of view). We define the subset of all possible PN's, producible by the 5 primitives and the 4 operators above defined, musical primitives, they constitute the basis of the MAP language. MAP makes possible the description of the structural level of music [De Poli and Haus 1982] as a multilevel environment within which musical objects may flow concurrently and interactively; the more detailed levels of representation (symbolic, physical, and operating levels) are provided by the CMUSIC package, which is automatically invoked by the MAP description. The final output is at the level of sound sample files. This approach extends the notion of musical process [Smoliar 1971] by the introduction of concurrent processes, without time explicit description as in [Dannenberg 1984); we show below how to introduce time description within Petri Nets. Hence, we can use PN musical primitives and operators to describe a musical text at a certain level of abstraction. Then, we can represent the same text at a lower or higher level of abstraction by means of suitable morphlsms. A mor- phism is a transformation from a PN "A" to a PN "B", which preserves the structure of a PN; an example of the application of a morphism from a complex musical text described by a PN is Figure 3. An example of the application of a morphism.

Describing and Performing Musical Processes 339 given in Fig.3; this example is an excerpt from the complete description of the Canon Perpetuus by J.S.Bach presented in [Degli Antoni and Haus 1982]. Thus, we can make a top-down description of musical texts, as suggested in [Qoguen 1975] and as in our purposes; we might also realize a bottom-up description as attempted in

[Bertoni et al. 1978] or a linguistic-systemic approach as in [Holtzman 1978]. The top-down approach is closer to the synthesis of texts (musical composition, syntax definition, etc.). Before we present the implementation of the MAP system in detail, let us recall two PN's, of particular relevance for musical description. To describe a cycle to be performed a certain number of times, we use a loop (see Fig.4); the place named CNT is a control object (a counter) which contains as many tokens as the number of cycles to be performed. To represent a logical/temporal dependence we use a WATT structure (see Fig.S), where SGN is a control object (a signal) which signals the occurrence of a musical object. Figure 4. Loop structure. SON y^ V / Figure 5. Wait structure.

 3. MUSICAL PROCESSES AS TIMED ENTITIES It has bean shown how a musical process can be described by a metaphor and a formal (graphic) language. The former has bean defined as a sat of autonomous "Objects", interacting via control "Tokens"; for the latter, the well known Petri Nets schema has been chosen, it Is worth noting that the two points (the metaphor and the language) are logically Independent. The metaphor emphasizes the distributed and concurrent nature of the musical process, which Is the result of the (possibly non deterministic) cooperation of several "actors", which communicate each other via "messages". Actors can perform at any level of abstraction; from a simple loop oontrol task to a "choral" or "symphonic" performance. In this scenario, the metaphor is very similar to Hewitt's Actor System for representing Knowledge [Hewitt et al. 1973] [Hewitt and Baker 1978], Including the characteristics of "openness" and "locality of knowledge* typical of (multiple) human cognitive and creative activities. The metaphor and its representing language are intended to formally describe the process, that is, analyze it to enhance the understanding of the process. To reproduce the process, or use its representation to synthesize an output rather than analyze its behavior, we have to jump from the Metaphor to a Model. We use this term to emphasize the computational characteristic (and thus the possibility of performing emulation runs) rather than a purely generic description. The model of a musical process is founded on a basic semantics, common to all musical objects: any object is a Timed Entity. Time is an Independent object that any other object Is forced to interact with. Objects act concurrently in time, and communicate/synchronize at particular instants on the time axis. The computational model must assure two conditions: 1) time

consistency, i.e., a unique time axis exists, 2) time resolution, I.e., two distinct points on the time axis can be adequately close. It la well known that Petri Nets are suitable to model "timed processes", and their use as Input language for the proposed system is straightforward. To this end, we have designed and implemented a proper linguistic (instead of graphic) formalism, and a virtual processor able to emulate the concurrent execution of a musical process modeled in this way. The language allows a "monodlmensional" linguistic description of a net, In terms of a sequence of statements. A Statement Sequence Is a full musical process, and is simply called "a net description". A statement may define: 1) a place, 2) a transition, 3) a particular marking, 4) starting and ending places of an execution. Symbolic names can be defined by the user, following simple rules, to designate objects. For example, the statement trana.t 0np1.sc Inp2.sc, out.net); Introduces a transition with the symbolic name trans with two Inputs and one output: Inputs are connected to Inpl and inp2, output is connected to out Suffix ac Indicates that symbols are CMUSIC score flies, suffix net Indicates that out Is a file containing a net description, and sufix t means transition.

Describing and Performing Musical Processes 341 Thus, a net description consists of two sets of statements: 1) a collection of statements (similar to that shown above) which define transitions, hence coping with the net topology, and 2) a collection of statements defining the starting and the ending places, and the Initial state of control tokens, hence coping with the net state. An example of the second kind of statements could be start placeO, cnt_one-2, cnt_two=5, sgn_71 -0; In this case the place "placeO" is assumed as the starting point, and Initial tokens are associated with places one, two, 71. To achieve a high level of simplicity, these are the only types of statements used. Furthermore, symbols (place and transition names) use a proper syntax to designate the type of object without requiring an additional statement: prefix cnt means Counter Token sgn means Signal Token suffix sc means CMUSIC score file net means Net Description File t means Transition default means Sound File Finally, the type of transition (or "musical primitive", as described in the previous section) is Implicitly defined by the number of inputs and outputs of a transition, since It can be identified unambiguously. Such a description of the language is obviously informal, and is only an attempt to Introduce the main features of it, without going into details; a complete description can be found in [Camurri 1984]. For the sake of clearness, figures

6a and 6b show a very simple net, together with its linguistic description. START ^SCORH loop OUT 2 "OUT1 Figure 6a. A simple example of a net.

342 A. Camurri, G. Haus and R. Zaccaria fus.t(cnt_start scorei.sc, outl.net out2); Ioop.t(o(jt2, soore1.sc); start scorel.sc, cnt_start>3; end outl.net; The execution starts at place score 1, and ends at outl; start has an Initial token value of 3. Figure 6b. Linguistic description of the net in figure 6a. A compiler has been designed (MFORM) which translates net descriptions Into a proper internal virtual code. Codes are executed by a virtual processor (PERFORM) which simulates the concurrent behavior and performance of the net model, whose typical output consists of sound sample files. Finally, a tool has also been implemented (REMARK) as an editing facility on the compiled code. The three programs will be discussed later on. Due to the concurrent and asynchronous nature of the execution process, some basic choices about the executing algorithm are necessary: such choices are briefly discussed In the next section. 4. NET EXECUTION Net execution is a distributed timed process. The problem of Implementing It on a single processor has a well known theoretical background [Dijkstra 1968, Brinch Hansen 1975, Hoare 1978). In our case, the execution process does not need to be performed In real time: this simplifies the task, since the two rules introduced in the previous section (consistency and resolution) must be satisfied in the overall output, and not at any particular execution Instant. We use Process as a common term to designate an active limed musical object associated with a place. In general, three main solutions can be devised: (1) Multiple Processes: in a multi - process environment (as Unix In this case), any place can activate a new process with which it will be associated; processes communicate through start/end events (in the Unix case, signals) and share common output structures (In the Unix case, multiplexed files, pipes or sockets). (2) Preemptive Scheduling, as In many simulation or concurrent languages, the central (virtual) processor is multiplexed among the processes: this function ie performed by a soheduier process (the Time Object) which guarantees simultaneous time-flow across all processes by giving a time slice to each in turn.

Describing and Performing Musical Processes 343 (3) Non-Preemptive Scheduling, this form of scheduling is similar to the previous, but in some ways easier; in fact, each place, once active, can use processor time as long as needed, until its own completion. This way, time flows in "bursts" and must be "folded back" when a process stops so the processor passes to another "simultaneous" process. The first solution is rather naive, though it may seem appealing at first glance. This is a general statement, which also applies to our situation, that is, under the Unix operating System1. The main reasons are: (1) The number of different processes that can be activated under a shell is limited and, for general use, small; (2) Owing to the kernel scheduling policy, the timing of concurrent processes is non deterministic, and even in optimal cases (e.g., single user) a strict parallelism cannot be achieved; (3) Communication primitives between processes under Unix are either tricky or inefficient if used at this level; for example: Standard Files need complicated and non-standard tricks to be used as shared resources (semaphore files, signals and so on); Multiplexed Files under 4.1 BSD have limited capabilities and are not supported under subsequent versions; Pipes are available in a very limited number; Signals are atomic mechanisms (events) and there is not full compatibility between Unix versions; Sockets under 4.2 BSD are very powerful, but basically intended for higher level communications, hence inefficient and limited in number for the present case. (4) System efficiency quickly decreases: processes related to places are typically disjoint, and a much more efficient execution is achieved if they are activated successively rather than simultaneously, without needlessly overworking the kernel's scheduler; furthermore, the use of communication tools, especially if designed for high level operations, would be another source of inefficiency. The peculiar nature of the net model permits the choice of a very efficient algorithm for net execution. In fact, we must take into account that processes are concurrent but disjoint, except for output sharing. However, we can separate the two problems by 1) using a non - preemptive scheduler which sequences the executions of processes without requiring time slicing and interruptions, and 2) providing a "collector" of all outputs, which makes the processes produce their own outputs separately, without requiring resource sharing for output channels. In other words, a net is executed by exploring recursively all its branches as deeply as possible, annotating the time flow involved; when a dead end is encountered, another branch is taken and so on; outputs are separately stored in different output channels together with their own "time axes", and, finally, the global output is reconstructed by mixing the partial outputs. 1 In this context, we reler to BSD Unix (4.1 and 4.2). However, no conceptual differences exist with other Unixes, except perhaps in minor details

 This algorithm will be described in detail in the next section. It is very efficient, and is also independent of the particular Unix version. It is also worth noting that it provides no short cut solution, and can successfully support a very high number of musical objects, as well as different levels of abstraction in the model. 5. IMPLEMENTATION ENVIRONMENT The MAP system has been implemented under the Unix environment: one of the reasons underlying this choice is that MAP has been designed as an extension of the representation capabilities of CMUSIC, the general purpose sound synthesis program developed at the Computer Audio Research Laboratory (CARL) of the University of San Diego, California, which operates under Unix. CMUSIC defines a language - whose overall design is based on the MUSIC V system - operating at a symbolic level of abstraction of the musical phenomenon, just like the traditional musical language, whereas the developed system can also operate at a higher level of abstraction, a structural one. To clarify this concept, let us examine how CMUSIC is connected and used In our system. The MAP system has been designed to describe and perform a Petri Net description (that we will simply call PN score), whose places are musical objects. In particular, a musical object may be, in our system, either a CMUSIC score, or a sound file containing samples in the CMUSIC output format, or a lower level PN. Furthermore, a place in a PN score may be a musical object as well as a control object. In the next section we shall examine in detail the appropriate software for description and performance of a PN score. It is worth noting that the execution utilizes the capability offered by Unix which makes it possible to call external programs or system commands from a process. This is useful, in particular, when the places of a PN refer to CMUSIC score files. The CMUSIC scores used as places in a PN score must observe some rules, In order to be correctly performed. In particular, they must have the same sample rate, output channel number, and other options which are in any case controlled by the system - by means of the Unix utility program grep - before calling the CMUSIC command. In developing the software (written in portable C) a wide use of Unix tools has been made, in particular: yacc, extremely useful for the implementation of compilers, and make, which provides a powerful mechanism for maintaining up-to-date versions of programs. 6. THREE PROGRAMS: MFORM, REMARK, PERFORM The MAP system for describing and performing musical processes has been implemented by a set of three user-level programs. The first - MFORM - defines a language for describing PN scores; the second - REMARK - provides some editing facilities to the user, and the

Describing and Performing Musical Processes 345 last - PERFORM - is the performer of PN scores processed by the previous two programs. Each program has a specific purpose in the system. MFORM requires that the graphic description of the input PN score has been converted into an equivalent linguistic one by the user. MFORM may be conceptually compared to the front end of a compiler: in fact, MFORM accepts a file containing a user level linguistic description of a PN score as input, and produces an of internal form, which is a lower level description of the given PN score. This internal form may subsequently be accepted as input by the REMARK and PERFORM programs. The input language has been chosen to meet several requirements. First, the language should be easy and evocative, second, it should not introduce constraints in the representation of PN scores, and, further, it should be easily modifiable, to adapt itself to possible future requirements. In fact, the system will probably grow during its use in musical composition. This last consideration is one of the reasons which led us to generate the parser of MFORM with the Unix tool yacc This makes it easier to modify and re-create the grammar as necessary. The actual version of the MFORM language has a context free LR(1) grammar. The internal form generated by MFORM is a list version of the PN score topology; it contains all information about the PN score suitable for net execution. In order to perform a PN score, another program - PERFORM - has been implemented, it accepts, as input, the internal form generated by MFORM, and outputs a set of sample files (output channels). We can take the net description given in fig. 6; the application of the PERFORM program gives an output channel structure as shown in fig. 7. Three output channels are represented on the time axis. The time duration of the score-places has been chosen arbitrarily. Note the files of pauses introduced into the channels which preserve the equality of time axes between channels. Of course, control places do not appear in the output channels, but control the performance flow; in the example, the counter START controls the number of iterations of the loop formed by score 1 and OUT2. SCORE1 OUT2 SCORE 1 OUT2 SCORE 1 OUT2 PAUSES OUT1 PAUSES OUT1 PAUSES OUT1 Figure 7. Output channels structure corresponding to the net of figure 6a.

346 A. Camurri, G. Haus and R. Zaccaria The algorithm followed by

PERFORM is a coveting algorithm of musical PN's, and can be briefly summarized as follows2: procedure performer (start place, time init); begin repeat if (start place is the final place of the score) then begin execute(start_place); return end; if (start place is an alternative input place) then begin < alternative management>; end; (* let us consider now the transition following the start place, which is uniquely determined by the alternative management, if an alternative does exist *) if (there is a fusion in such a transition) then < fusion management > else execute(start_place); If (there is a splitting in such a transition) then < recursive call to performer for each splitting branch in output to the transition (the first, next_place, excepted) > start_place : = next_place; end until (score performed); < management of deferred alternatives >; end C performer •) This is a recursive algorithm, which entirely covers the PN, beginning at its starting place, performs the musical places, synchronizes them, and sends the outputs to the sample files (output channels). We have seen that a place in a PN score may be a sound Tile, a CMUSIC score, a signal or a counter: it may, in its turn, be an internal form of a lower level PN score. This capability is made possible by the implementation of the MAP system as a set of sequential programs: this capability permits the abstraction, at a higher structural level, of a description of musical processes. In this case, the PERFORM algorithm performs the lower level net recursively. In the other cases, the performer algorithm when performing a place behaves in the following way: i) If it is a sound file, it is put into an appropriate output channel with the proper timing (see fig. 6); ii) if the place is a CMUSIC score, the PERFORM algorithm calls the appropriate system command to perform the score by the CMUSIC system (the sound file thus generated is subsequently sent to an output channel, according to its timing); iii) a place which is a signal or a counter produces no performance, but it serves to control the performance flow. It is worth noting the particular management of the musical primitives fusion and alternative, which are not specified in the previous brief description of the algorithm. For this purpose we will use a simplilled PASCAL - like formalism.

 Let us analyze first the fusion management. Every time the PERFORM algorithm meets a transition in which there is a fusion, it checks that it has already processed all the other inputs related to the other branches; if so, the fusion is called ready fusion. If it is a ready fusion, all the input places are performed and sent, with appropriate synchronization, to the suitable output channels. Otherwise, if the fusion is not yet ready (i.e., if the program has not yet covered the relative branches), the performance stops in this branch, marking as processed the last place it has met, resuming the execution of another eligible branch of the PN; during the execution of the rest of the net, there will be a time when the suspended fusion becomes ready3. Now let us analyze how the musical primitive alternative is managed. The philosophy of the algorithm is the same for fusion, as in any case PERFORM is asked to execute a set of parallel activities - the flow of the tokens in the PN score - in a sequential way. In the case of alternative, PERFORM has to check several things before deciding its behavior. Let us consider the example in Fig.8, in which there is one alternative (in Pall) and two fusions. We suppose the performer hes arrived in Pall, the input place of the alternative. The algorithm may behave in different ways, according to the state of the places P1, P2 and Palt. The state of a place is defined as the number of tokens and the timing associated to each token. According to the place state, the algorithm gees on, choosing the branch toward t1 or 12, or it stops, if it has not yet examined the state of PI and/or P2. To this end, we must introduce two definitions: by performance timing of a token, we mean the simulated time associated with that token, that is, the time axis that will correspond to the real time performance of the sound samples generated by the token flow; by execution timing we mean the instant of time when, during the execution of the PERFORM algorithm, a token reaches a particular place or transition. Let us analyze in detail the behavior of the algorithm in the example. Suppose it has already processed the branches ending in PI and P2 (otherwise the alternative management P1 _Palt ^P2 O O Figure 8. Example of a simple alternative - fusion block. 1 This excludes a priori the possibility ot a deadlock in the PN.

 would be deferred), that is, at least one token is present in both Pi and P2 at the execution time et at which a token arrives in Palt. If we name t(P7), %P2) and t(Pa/r) the respective performance timings of PI, P2 and Palt, the algorithm behaves in one of the following ways, according to the state of the places: (1) Non-deterministic behavior: if %P1) and t(P2) are less than (or equal to) t(Pa/r), PERFORM makes a random choice between the branches which lead to t1 and t2. (2) Deterministic behavior: if t(P7) is less than (or equal to) t(Pa/l), and \{P2) is greater than t(Pafl), the algorithm chooses the branch leading to t1. (3) Stopped alternative:

if t(P7) and t{P2) are both greater than t(Pa/r), then the alternative will stay blocked in this branch for ever. It is also possible that P1 or P2 or both never receive any token. This is the same as having the respective performance times at infinity. In this and other particular cases, the alternative management is deferred to the end of the PERFORM algorithm. (4) Lastly, if either Pi or P2, or both, have no token at et, execution stops and another branch of the net resumes; the alternative will be performed later, as soon as the missing token P1 or P2 is available. The behavior will be the same as that described above, i.e., in terms of performance timing (which determines the semantics) instead of execution timing (which determines the scheduling). The example in Fig.6 shows one alternative and two fusions: this "fusion - alternative - fusion cell" is the simplest structure which exhibits the non-deterministic behavior discussed above. However, more general structures based on composition of such elementary cells or on alternative and fusion with more than two branches are supported by MAP (MF, or Macro Fusion blocks). For the sake of briefness we will not describe this point in detail. The user can keep the flow of the program PERFORM under control by means of a "verbose" option, which causes the printing on the terminal of a detailed description of program activities. Other options are available to allow partial performance of the given PN. The third program - REMARK - is an interactive internal form editor. REMARK makes possible the reading and the editing of an internal form created by MFORM. The possible modifications are alterations of the initial marking, or exchanges of places in the PN score. Changes cannot be made in the topology of the net. In other words, REMARK provides a definition of a class of internal forms which all have the same topology and derive from the internal form of the original PN score. REMARK is designed to give further facilities to a user of the MAP system, and avoids the re - compilation by MFORM of a PN score in the case of certain common modifications. These three programs do not put particular limitations on the net parameters. In particular, there are no implementation limits on the number of links between places and transitions (for instance, we may have fusions with more than two input places).

 For each program there is a source file containing ell the implementation modifiable parameters, i.e., the maximum dimension of the PN score, or the maximum number of links. Unix - format user manuals have been written for these programs. As a last consideration, it is worth observing that the final output produced by the MAP system - the set of channels - requires a further mixing phase. This phase is conceptually disjoint from the system, and is very important for real musical applications of the MAP system. This can be done either digitally or analogically. The former requires the software implementation of a mixing console operating in real time, with the advantages deriving from the digital processing, such as the capability to edit directly on sound samples. The latter is closer to the traditional approach, which basically consists of an analog mixing console and a multi-track tape recorder. The output channels produced by the MAP system can be recorded on separate synchronized tracks and then mixed with the subsequent well known advantages and limitations.

 APPENDIX A: A Fugue by J.S. Bach As a musical example, we have analyzed the first part of the XXIII fugue from "The Well Tempered Clavichord" (vol.1), by J.S.Bach (see Fig.9). It consists of the exposition of a tonal, four part fugue, characterized by two counter - subjects: this means that there are four distinct parts, whose entries are in a canon style, each beginning with the Theme (always marked with the letter T, in the score of Fig.9), followed by the two counter-subjects (a sort of "answers" to the Theme), except for the entry of the last part, which concludes the fugal exposition. The four occurrences of the Theme can be summarized in the PN of Fig.10, particularly at places T1, T2, T3 and T4. The first time the PN is activated, T1 is executed (which corresponds to bars 1 and 2 in the score), and T2 becomes avaiable for execution (a token is placed in SQN_T2). So, the next activation of this PN will elicit the execution of T2 (bars 3 and 4 of the contralto part), and so on for T3 and T4. We can use this PN as a place in a higher level PN which represents the entire fugal exposition. For Instance, the PN in Fig.11 is one of the possible ways to do It. The places of such a PN are, obviously, either control places or fragments of the musical score of Fig.10, i.e. musical objects. In particular, the places named THEME1 and THEME2 are instances of the previously analyzed net, which represent the various Theme entries. Likewise, CS1 and CS2 represent the occurrences of the two counter - subjects, which can be described In the same Fuga XXIII Figure 9. Score of the exposition of the XXIII fugue from the "Well Tempered Clavicord" (vol. 1) by J.S. Bach.

 I TART Q- END Figure 10.

PN summarizing the four occurrences of the Theme in the fugue. way as the Theme in THEME1 and THEME2. We have used in the score two "dummy" names, THEME1 and THEME2, to obtain two distinct places both referring to the same lower level PN "T.net". The place CODA represents bars 7 and 8 of the tenor part, which has already exposed the Theme and its oounter-subjects. The fugal exposition terminates et the beginning of bar 9. This is basically an analysis of the musical text. If we go on to the synthesis we can consider the PN scores taken from the score of the fugue and imagine them performed by the MAP system. A description of the PN representing the entire score (Fig. 11) can be converted into the following MAP-score: start SQN_start, CNT » 3, SGN_CODA - -2, SQN_CS2 - -1, CNT out - -4; tl.t (SQN_start, $THEME1); Figure 11. One of the possible PN's corresponding to the score of figure 9.

352 A. Camurri, G. Haus and R. Zaccaria t2.t($THEMEl,CNTSGN); t3.t (CNT SGN, SGN CODA SGN_CS2 CS1.net $THEME2); t4.t (SGN_CODA, c5bA.sc); t5.t (SGN CS2, CS2.net); t6.t (CODATsc CS2.net CS1.net STHEME2 , CNT_out SGN); t7.t (CNT_out, SGN_END); end SGN_END; T.net = $THEME1 = $THEME2; The place CODA is taken to be a CMUSIC score, whereas CS1, CS2, THEME1 and THEME2 are lower level PN's, as we have seen.4 Fig. 12. The output channels (sample files) generated by the PERFORM program are shown Tl CS11 CS12 CODA PAUSES T2 CS21 CS22 PAUSES T3 CS13 : : • PAUSES T4 Figure 12. Output channels generated by PERFORM. 4 MAP scores ol these nets are not given here

Describing and Performing Musical Processes 353 APPENDIX B: Petri Nets A Petri Net (PN) is a graph which may have two types of nodes: a) places, represented by circles; b) transitions, represented by bars. Nodes may be connected by directed arcs places may be connected to (and only to) transitions; transitions may be connected to (and only to) places. If an arc is directed from a place (transition) to a transition (place), then the place (transition) is an input to the transition (place) and the transition (place) is an output to the place (transition). Places may contain tokens, represented by markers. A PN with tokens is a marked PN; tokens are moved by the firing of the transitions of the net; a transition is enabled to fire when all of its input places have (at least) one token. A transition fires by removing the enabling tokens from its input places (one token for each input place) and generating new tokens which are put in the output places (one token for each output place). The distribution of tokens in a marked PN is called marking, and the initial state of a PN is its initial marking. The major features of PN's are: asynchrony, nondeterminism, concurrency, causality and hierarchy. Furthermore, PN's allow to describe "conflict structures" and time - dependent relationships. In general, systems modeled by PN's may be deterministic or not, synchronous or asynchronous. A typical characteristic of a PN is that it stands on small basic set of primitive structures suitable for the representation of processes and their interactions. We may have two kinds of concurrency within a PN: concurrency without conflict (see Fig. 13 and Fig. 15) and concurrency with conflict (see Fig. 14). In Fig. 13 the two enabled transitions can fire with no interaction; in Fig. 14 there are two enabled transitions in conflict (only one transition fires and disables the other); a second token in place B (Fig. 15) causes a situation similar to the previous, hence place B can carry on a control function about the number of firings. From another point of view, Fig. 13 shows two transitions which can fire simultaneously, while Fig. 14 shows a PN in which transitions cannot fire simultaneously. '© © I-- Figure 13. Independent processes.

 Figure 14. Conflict structure (alternative). Figure 15. Control structure. Figure 16. Wait structure.

 In Fig. 16, the place C has a function similar to a wait construct: transition Y can fire if and only if transition X has already fired. In this way concurrent processes can be synchronized where it is required. From a general point of view we can describe the interactions among several processes by means of a minimal set of PN primitive structures combined in various ways. In Fig. 17 eight structures are shown; their meaning is formally described as follows: 17a: (process X xor process Y) enabled by A; 17b: (process X and process Y) enabled by A; 17c: process X enabled by (A and B); 17d: process X enabled by (A or B); 17e: A produced by (process X or process Y); 17f: A produced by (process X and process Y); 17g: (A andB) produced by process X; 17h: (A xor B) produced by process X. From the point of view of processes, Fig. 17a represents the non deterministic alternative, Fig. 17b represents concurrency, Fig.17e represents asynchrony, Fig.17f represents synchrony. Fig. 13, on the other hand, represents independent processes. From the point of view of resources, Fig. 17c represents strong necessity, Rg.17d represents weak necessity, Fig.17g represents joint production, Fig.17h represents alternative production. Literature about Net Theory started in the early sixties with a monography by Petri [1982a]i, who further investigated the matter in other papers [1963b, 1978, 1977, 1978 etc.). A detailed bibliography on the subject is in Pless [1980] and textbooks about PN's are now avaiable, for example Peterson [1981] and Reisig [1983] - to mention just a couple of them. Finally, we wish to mention a set of papers that specifically deal with the issue of describing concurrency in PN's: Noe and Nut [1973], Ramchandani [1974], Petri [1978], Herzog [1980], Ramamoorthy and Ho [1980], Nielsen et al. [1981], Magott [1984].

A. Camurri, G. Haus and R. Zaccaria e Figure 17. Basic structures.

357 References ABEND W, BIZZI E, and MORASSO P (1982) Human Arm Trajectory Formation. Brain 42:331 - 338. ADAMOVICH SV, and FELDMAN AQ (1984) Model for Central Parametrization of Movement Trajectories. Biofizika 29: 309-312. ADORNI Q, GAQLIO S, and ZACCARIA R (1964) Towards a Description of Robot Movements by a Quasi - Natural Language. Theoretical Linguistics. 11:45-89. AGARWAL GC, and GOTTLIEB GL (1977) Compliance of the Human Ankle Joint. J. Biomech. Eng. 99: 166-170. ALEKSANDER I (1978) The Human Machine. Georgi Publisher:St Saphorin, Switzerland. ALEKSANDER I, and BURNETT P (1983) Reinventing Man. Penguin Books:London. ALEKSANDER I (1983) Emergent Intelligence from Pattern Recognition. In: Artificial Vision for Robots (I. Aleksander, Editor). Kogan Page:London. ALLEN P, and BAJCSY R (1965) Object Recognition Using Vision and Touch. Proceed. IJCAI- 85 (Los Angeles, California August 18-23): 1131 -1135. ANDERSON JR, and BOWER QH (1973) Human Associative Memory. VH Winston and Sons:New York. ANDERSON JR (1976) Language, Memory, and Thought. L. Erlbaum Assoc.:Hillsdale, New York. ANDREWS QR, and SCHNEIDER FB (1983) Concepts and Notations for Concurrent Programming. ACM Computing Surverys 15:3-43. ANDRIACCHI TP, OQLE JA, and QALANTE JO (1977) Walking Speed as a Basis of Normal and Abnormal Qait Measurement. J. Biomedicine 10:261. ARBIB MA (1981) Perceptual Structures and Distributed Motor Control. In: Handbook of Physiology - The motor system II. Motor control. (V. Brook, Editor). American Physiological Soc.:Bethesda, Maryland. ASADA H (1983) A Geometrical Representation of Manipulator Dynamics and its Application to

Arm Design. Trans. ASME J. Dynamic System Measurement Control 105: 131 -136. ASADA H, and BRADY M (1984) The Curvature Primal Sketch. Proceed. IEEE Workshop Computer Vision - Representation and Control (Annapolis Maryland, April 30-May 2):8- 17. ASATRYAN DG, and FELDMAN AQ (1965) Functional Tuning of Nervous System with Control of Movement or Maintenance of a Steady Posture. I. Mechanographlc Analysis of the Work of the Joint on Execution of a Postural Task. Biofizika 10: 837-846. BADLER Nl, and SMOLIAR SW (1979) Digital Representation of Human Movement. Computing Surveys 11:19-38. BADLER N, guest editor (1983) Modeling the Human Body for Animation. Special Issue IEEE J. Computer Graphics and Applications 2(9). BALLARD DH, and BROWN CM (1982) Computer Vision. Prentice Hall:Englewood Cliffs.N.J.

 BARBER C (1962) Office Automation Semantics. Ph.D. Thesis, MIT, Cambridge, Massachusetts. BARBER C, DE JONG P, HEWITT C (1963) Semantic Support for Work in Organizations. MIT- Al Laboratory Memo 719. BARR AM (1981) Superquadrics and Angle Preserving Transformations. IEEE J. Computer Graphics and Applications 1:11 -23. BARROW HG, and TENENBAUM JM (1981) Computational Vision. IEEE Proceed. 69:572-595. BARSKY BA, and GREENBERG DP (1980) Determining a Set of B- Spline Control Vertices to Generate an Interpolating Surface. Computer Graphics and Image Processing 14:203 -226. BARTENIEFF I, and LEWIS D (1960) Body Movement: Coping with the Environment. Gordon and Breach: New York. BAUMGART BG (1974) Geometric Modeling for Computer Vision. Ph.D. Thesis. Stanford Univ., Stanford, California. BAYNES CF (1951) The I Ching: or the Book of Changes. Routledge and Kegan Paul: London. BELL CG (1985) Multis: A New Class of Multiprocessor Computers. Science 228:462-467. BENATI M, GAGLIO S, MORASSO P, TAGLIASCO V, and R. ZACCARIA (1980) Anthropomorphic Robotics. I: Representing Mechanical Complexity. II: Analysis of Manipulator Dynamics and the Output Motor Impedance. Biological Cybernetics 36:125-150. BENATI M, MORASSO P, and TAGLIASCO V (1982) The Inverse Kinematic Problem for Anthropomorphic Manipulator Arms. Trans. ASME J. Dynam. Systems Measurement Control 104:110- 113. BEN-ISRAEL A, and GREVILLE TN (1974) Generalized Inverses: Theory and Applications. John Wiley and Sons:New York. BERKINBLIT MB, FELDMAN AG, and FUKSON Ol (1985) Adaptability of

Innate Motor Patterns and Motor Control Mechanisms. The Behavioral and Brain Sciences. BERKLING KJ (1971) A Computing Machine Based on Tree Structures. IEEE Trans. Computer C-20:404-418. BERNSTEIN NA (1967) The Coordination and Regulation of Movements. Pergamon Press:London. BERTONI A, HAUS G, MAURI G, and TORELLI M (1978) A Mathematical Model for Analyzing and Structuring Musical Texts. Interface 7. BESL PJ, and JAIN RC (1985a) Three - Dimensional Object Recognition. ACM Computing Surveys 17:75-145. BESL PJ, and JAIN RC (1985b) Invariant Surface Characteristics for Three - Dimensional Object Recognition in Range Images. Computer Vision, Graphics and Image Processing. BEZIER PE (1972) Numerical Control: Mathematics and Applications. John Wlley:London. BINFORD TO (1982) Survey of Model Based Image Analysis Systems. Intern. J. Robotics Res. 1:16-64.

 BISSHOP KE (1969) Rodrigues Formula and the Screw Matrix. Trans. ASME J. Engineering Industry 91:179-185. BIZZI E, KALIL R, MORASSO P, and TAGLIASCO V (1972) Central Programming and Peripheral Feedback during Eye - Head Coordination in Monkeys. In: Central Control of Eye Movements and Motion Perception (J. Dichgens and E. Bizzi, Editors):220-232. KargenBasel. BIZZI E, POLIT A, and MORASSO P (1976) Mechanisms Underlying Achievement of Final Head Position. J. Neurophysiology. 39:435-444. BIZZI E, DEV P, MORASSO P, and POLIT A (1978) Effect of Load Disturbances During Centrally Initiated Movements. J. Neurophysiology 41:542-556. BIZZI E, ACCORNERO N, CHAPPLE W, and HOGAN N (1982) Arm Trajectory Formation in Monkeys. Exper. Brain Res. 46:139-143. BIZZI E, CHAPPLE W, and HOGAN N (1982) Mechanical Properties of Muscles: Implication for Motor Control. Trends in Neuroscience 5: 395-398. BLUM H (1973) Biological Shapes and Visual Science. J. Theoretical Biology 38:205-267. BLUM H, and NAGEL RN (1978) Shape Description Using Weighted Symmetric Axis Features. Pattern Recognition 10:167-180. BOBATH B, and BOBATH K (1975) Motor Development in the Different Types of Cerebral Palsy. W. Heneman Medical Books Ltd:London. BOBBERT AC (1960) Energy Expenditure in Level and Grade Walking. J. Applied Physiology 15:1015. BOBROW DG (1984) Qualitative Reasoning about Physical Systems: an Introduction. Artificial Intelligence 24:1-6. BOCCARDI S, and PEDOTTI A (1982) Valutazione Sperimentale del Passo in Funzione Riabilita- tiva. Proceed. 2nd ANFFAS National Congress (Casale

Monferrato):l7-29. BOLLES RC, and CAIN RA (1982) Recognizing and Locating Partially Visible Objects: the Local-Feature-Focus Method. Intern. J. Robotics Res. 1:57-82. BOLLES RC, HORAUD P, and HANNAH MJ (1983) 3DPO:A Three - Dimensional Part Orientation System. Proceed. IJCAI-63 (Karlsruhe, August 8-l2):1H6-1120. BONNER S, and SHIN KG (1982) A Comparative Study of Robot Languages. IEEE Computer 15:12. BOOYENS J, and KEATINGS WR (1957) The Expenditure of Energy by Men and Women Walking. J. Physiology 138:165. BRADY M, HOLLERBACH JM, JOHNSON TL, LOZANO-PEREZ T, and MASON M (1982) Robot Motion: Planning and Control. MIT Press:Cambridge Massachussets. BRADY M, and ASADA H (1984) Smoothed Local Symmetries and their Implementation. Int. J. of Robotics Res. 3:36-61. BRAIN L, and WALTON JN (1989) Diseases of the Nervous System. Oxford Univ. Press:London. BRIDWELL NJ, and HUANG TS (1983) A Discrete Spatial Representation for Lateral Motion

 Stereo. Computer Vision Graphics and Image Processing 21:33-57. BRILLOUIN L (1964) Tensors in Mechanics and Elasticity. Academic Press: New York. BRINCH HANSEN P (1975) The Programming Language Concurrent PASCAL. IEEE Trans. Software Engineering SE-1:199-206. BRINCH HANSEN P (1977) The Architecture of Computer Programs. Prentice HalkEnglewood Cliffs, New Jersey. BROOKS VB, COOKE JD, and THOMAS JS (1973) The Continuity of Movements. In: Control of Posture and Locomotion (R.B. Stein, K.Q. Pearson, R.S. Smith, and J.B. Bedford, Editors):257- 272. Plenum Press: New York. BROOKS RA (1982) Symbolic Error Analysis and Robot Planning. Intern. J. Robotics Res. 1:29-68. BROOKS RA (1983) Model-Based Three - Dimensional Interpretations of Two - Dimensional Images. IEEE Trans. Pattern Analysis Machine Intelligence PAMI-5:140-149. BROWN MD, SMOLIAR SW, and WEBER L, (1976) A Graphic Editor for Labanotation. Computer Graphics 10: 60-65. BROWN CM (1982) PADL-2: A Technical Summary. IEEE J. Computer Graphics and Applications 2:69-84. BRUNNSTROMM S (1954) Center of Gravity in Relation to Ankle Joint in Erect Standing: Application to Posture Training and Artificial Legs. Physical Therapy Review 34:109. BRYAN WL (1892) On the Development of Voluntary Motor Ability. American J. Psychology 125. BUDAK BM, and FOMIN SV (1973) Multiple Integrals, Field Theory and Series. Mir Publishers: Moscow. CALVERT TW, and CHAPMAN J, PATLA A (1982) Aspects of the Kinematic Simulation of Human Movements. IEEE J.

Computer Graphics and Applications 2:41 - 50. CAMURRI A (1984) Un Linguaggio per la Descrizione e la Esecuzione di Processi Musicali Basato sulle Reti di Petri. Master Thesis, DIST, Univ. of Genoa, Genoa. CANNY J (1984) On Detecting Collisions Between Polyhedra. In: Proceed. ECAI - 84:533 - 542. Elsevier Science Publisher:Amsterdam. CARLBOM I, and PACIOREK J (1978) Planar Geometric Projections and Geometric Transformations. ACM Computing Surveys 10:465-502. CARPENTER RHS (1977) Movements of the Eyes. Pion: London. CAVAGNA GA, and KANEKO M (1977) Mechanical Work and Efficiency In Level Walking and Running. J. Physiology: 268 - 467. CHIANG Y (1973) Chinese Calligraphy. Harvard Univ. Press: Cambridge, Massachussets. CHOMSKY N (1957) Syntactic Structures. Mouton:The Hague, The Netherlands. CLINGER (1983) The Semantics of Actor Systems. Ph.D. Thesis, MIT, Cambridge, Massachusetts.

COHEN PR, FEIQENBAUM EA (1983) The Handbook of Artificial Intelligence. W. Kauffman:Los Altos, California. CONNOLLY CL (1985) The Determination of the Next Best Views. In: Proceed. IEEE Intern. Conf. Robotics and Automation (St. Louis Missouri, Mar. 25-28):435-435. CONRAD B, BENECKE R, CARNEHL J, HOHNE J, and MEINCK HM (1983) PathophysioJogical Aspects of Human Locomotion. In: Motor Control Mechanisms in Health and Disease (J.E. Desmedt, Editor) - Advances in Neurology 39:717-736. Raven Press:New York. COOKE JD (1980) The Organization of Simple Skilled Movements. In: Tutorials in Motor Behavior (Q.E. Stelmach, J. Requin, Editors): 199-201. North Holland:Amsterdam. COONS SA (1974) Surface Patches and B- Spline Surfaces. In: Computer Aided Design (Barnhill and Riesenfeld, Editors). Academic Press:New York. COX MO (1973) The Numerical Evaluation of B- Splines. J. Inst. Math. Applic. 10:134-149. CRAQO PE, HOUK JC, and HASAN Z (1976) Regulatory Actions of the Human Stretch Reflex. J. Neurophysiology 39:925 - 935. CURRY HB, and SCHOENBERQ IJ (1986) On Polya Frequency Functions IV: The Fundamental Spline Functions and Their Limits. J. Analyse Mathem. 17:71 -107. DAHL M, and NYQAARD K (1968) The SIMULA 67 Common Base Language. Norwegian Computing Center. DANNENBERQ RB (1984) A Functional Language for Real Time Control. Communications ACM 27:6. DEAN QA (1965) An Analysis of the Energy Expenditure in Level and Grade Walking. Ergonomics 8:31. DE BOOR C (1972) On Calculating with B-Splines. J. Approx. Theory 6: 50-62. DEQLI ANTONI Q, and HAUS Q (1982)

Music and Causality. Proceed. Intern. Computer Music Association (San Francisco, California). DE KLEER J (1975) Qualitative and Quantitative Knowledge in Classical Mechanics. MIT-AI Laboratory Technical Report 352, Cambridge, Massachussets. DE KLEER J, and BROWN JS (1984) A Qualitative Physics Based on Confluences. Artificial Intelligence 24:7-83. DE POLI Q, and HAUS Q (1982) Ingegneria del Software e Informatica Musicale. Proceed. AICA Congress. Univ. of Padova, Padova. DENAVIT J, and HARTENBERQ RS (1955) A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. J. Applied Mathematics 77:215-221. DENNIS JB (1980) Data-Flow Supercomputers. IEEE Computer 13:48-58. DESSIMOZ JD (1980) Traltement des Contours en Reconnaissance des Formes Visuelles: Application en Robotique. These n. 387. Ecole Polytecnique de Lausanne, Lausanne. DICHQANS J, BIZZI E, MORASSO P, and TAGLIASCO V (1973) Mechanisms Underlying Recovery of Eye Head Coordination following Bilateral Labyrinthectomy In Monkeys. Exp. Brain

362 References Res. 18:548-562. DUKSTRA EW (1968) Cooperating Sequential Processes. In: Programming Languages (F.Qenuys Editor). Academic Press:New York. DOOLEY M (1982) Anthropometric Modeling Programs - A Survey. IEEE J. Computer Qraphics and Applications 2:17-25. ORAKE.SH (1977) Using Compliance in Lieu of Sensory Feedback for Automatic Assembly. J. of Neurophysiology 39: 925-935. ELLIS CA, and NUTT QJ (1980) Computer Science and Office Information Systems. ACM Computing Surveys 12:27-80. ESHKOL N, and WACHMAN A (1958) Movement Notation. Weldenfeld & Nicholson:London. FAHLMAN SE (1974) A Planning System for Robot Construction Tasks. Artlfloial Intelligence 5:1-49. FAUQERAS OD, HEBERT M, PAUCHON E, and PONCE J (1984) Object Representation, Identification and Positioning from Range Data. NATO ASI series F11:255 -277. FAUX IP, and PRATT MJ (1979) Computational Geometry for Design and Manufacture. Ellis Hor- wood (J. Wiley): Chichester. FELDMAN AQ (1966) Functional Tuning of the Nervous System with Control of Movement or Maintenance of a Steady Posture. II. Controllable Parameters of the Muscles. III. Mechanographic Analysis of the Execution by Man of the Simplest Motor Tasks. Biophysics 11:565 -578;768-775. FELDMAN AQ (1974) Control of the Length of a Muscle. Bioflzlka 19: 749-753. FELDMAN AQ (1980) Superposition of Motor Programs. Neuroscience 5: 81 -95. FIKES RE, and NILSSON NJ (1971) STRIPS: A

New Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence 2:189-208. FITTS PM (1954) The Information Capacity of the Human Motor System In Controlling the Amplitude of Movements. J. Exper. Psychology 47:381-391. FLANAGAN JL (1982) Talking with Computers: Synthesis and Recognition of Speech by Machine. IEEE Trans. Biomedical Engineering BME- 29:223 -232. FLASH T, and MUSSA IVALDI FA (1984) Inferring Movement and Muscle Synergies from Multi- Joint Arm Posture. Abstract 14th Annual Conf. Society Neuroscience (Anaheim, California, Oct. 10-15):635. FOLEY JD, and Van DAM A (1983) Fundamentals of Interactive Computer Qraphics. Addison Wesley Publishing Co. FODOR JA (1976) The Language of Thought. Harvester Presa:Hassocks, Sussex. FORBUS KD (1984) Qualitative Process Theory. Artlfloial Intelligence 24:84-168. FRANKEL VH, and NORDIN M (1960) Basic Biomechanics of the Skeletal System. Lea & Febiger: Philadelphia.

 FU KS (1971) Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control. IEEE Trans. Automatic Control AC- 16:70-72. FLINT BV (1983) Analogical Modes of Reasoning and Process Modeling. IEEE Computer 16:10. 99-104. QAQLIO S, MASSONE L, and MORASSO P (1983) Representing Shapes for Visuo- Motor Processes in Robots. Proceed. IFAC Intern. Symp. Artificial Intelligence (Leningrad, October 4 - 6). QAQLIO S, MARINO Q, and ZACCARIA R (1965) Logic Programming on Reduction/Dataflow Machines. In: Artificial Intelligence - Methodology, Systems, Applications (W. Bibel and B. Petk- otf, Editors):23l - 239. North - Holland:Amsterdam. QALLISTEL CR (1981) The Organization of Action: A New Synthesis. Behavioral and Brain Science 4: 609 - 650. QIBSON JJ (1979) The Ecological Approach to Visual Perception. Hougton Mifflin:Boston. GILBERT EQ, and JOHNSON DW (1985) Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles. IEEE J. of Robotics and Automation RA-1:21 -30. QOERTZ RC (1963) Manipulators Used for Handling Radioactive Materials. In: Human Factors in Technology (E.M. Bennet, Editor). McGraw-Hill, New York. QOQUEN JA (1975) Complexity of Hierarchically Organized Systems and the Structure of Musical Experience. UCLA C.S. Dept. Quarterly:3(4). QOLDBERQ A, and ROBSON D (1983) SMALLTALK- 60: The Language and its Implementation. Addison - Wesley: Boston. QOLDBERQ AA, BENHABIB B, and FENTON

RQ (1985) A Complete Generalized Solution to the Inverse Kinematics of Robots. IEEE J. of Robotics and Automation RA-1:14-20. GOLDSTEIN H (1965) Classical Mechanics. Addison -Wesley:Reading, Massachusetts. QOLDWASSER SM (1984) A Generalized Object Display Processor Architecture. IEEE SIQARCH Newsletter 12:38-47. QORDON AM, HUXLEY A, and JULIAN FJ (1966) The Variation in Isometric Tension with Sarcomere Length in Vertebrate Muscle Fibers. J. Physiology 184:170-192. QORDON WJ, and RIESENFELD DP (1974) B-Spline Curves and Surfaces. In: Computer Aided Design (Barnhill and Riesenfeld, Editors). Academic Press:New York. GRAUSTEIN WC (1947) Differential Geometry. Princeton Univ. Press. GRILLNER S (1972) The Role of Muscle Stiffness in Meeting the Changing Postural and Locomotor Requirements for Force Development by Ankle Extensors. Acta Physiologica Scandinavica 86:92-108. QRILLNER S (1975). Locomotion in Vertebrates: Central Mechanisms and Reflex Interactions. Physiological Review 55:247 - 304. GRILLNER S (1981) Control and Locomotion in Bipeds, Tetrapods and Fish. In: Handbook of Physiology - Motor Control (V.B. Brook, Editor) 111:1179-1236.

 QRIMM RJ (1963) Program Disorders of Movement. In: Motor Control Mechanisms in Health and Disease (J.E. Desmedt, Editor) - Advances in Neurology 39:1-12. Raven Press:New York. QRIMSON WEL (1981) From Images to Surfaces. MIT Press:Cambridge, Massachussets. GUTTAQ J (1977) Abstract Data Types and the Development of Data Structures. Communications of the ACM 20:396-404. HALL EL, TIO JBK, MCPHERSON CA, and SADJADI FA (1982) Measuring Curved Surfaces for Robot Vision. IEEE Computer 15:42-54. HANAFUSA H, and ASADAA H (1977) Stable Prehension by a Robot Hand with Elastic Fingers. Proceed. 7th Int. Symp. on Industrial Robots. (Tokyo):36l - 368. HARALICK RM, LAFFEY TJ, and WATSON LT (1983) The Topographic Primal Sketch. Intern. J. Robotics Res. 2:50-72. HARRIES JO (1983) Language of Shape and Movement. The Movement Notation Society:Tel Aviv. HATZE H (1977) A Complete Set of Control Equations for the Human Musculo - Skeletal System. J. Biomechanics 10:799-805. HAYNES LS, LAN RL, SIEQUIOREK DP, and MIZELL DW (1982) A Survey of Highly Parallel Computing. IEEE Computer 15:9-26. HAYES PJ (1979) The Naive Physics Manifesto. In: Expert Systems and Microelectronic Age i(D. Michie, Editor). Edinburgh Univ. Press:Edinburgh. HELLEBRANDT FA, and FRIES EC (1942) Constancy of the Oscillographic Stance Patterns. Physical Therapy Rev. 121:471. HERBISON EVANS D (1982) Real-Time Animation of Human Figure Drawings with Hidden - Lines Omitted. IEEE J. Computer Graphics and Applications 2:27-33. HERMANN GT (1980) Image Reconstruction from Projections, the Fundamentals of Computerized Tomography. Academic Press:New York. HERZOG O (1960) Static Analysis of Concurrent Processes for Dynamic Properties using Petri Nets. Lecture notes in Computer Science:70. Springer Verlag:Berlin. HEWITT C (1969) PLANNER: a Language for Proving Theorems in Robotics. IJCAI - 69 (Washington DC):295-302. HEWITT C (1972) PLANNER. MIT-AI Laboratory Technical Report 258, Cambridge, Massachusetts. HEWITT C, BISHOP P, and STEIGER R (1973) A Universal Modular Actor Formalism for Artificial Intelligence. Proceed. IJCAI - 80:235 - 245. HEWITT C, and SMITH B (1975) Towards a Programming Apprentice. IEEE Trans. Software Engineering SE -1:26 - 45. HEWITT C, and BAKER H (1978) Actors and Continuous Functlonals. In: Formal Description of Programming Concepts (Neuhold Editor). North Holland:Amsterdam. HEWITT C (1960) The Apiary Network Architecture for Knowledgeable Systems. Proceed. LISP

 Conf. (Stanford, California):i07-118. HEWITT C, and DFE JONG P (1983) Open Systems. In: Perspectives on Conceptual Modeling (M.I. Brodie and J.L. Mylopoulos, Editors). Springer Verlag:Berlin. HILLIS WD (1985) The Connection Machine. MIT Press:Cambridge, Massachusetts. HINTON G (1984) Some Computational Solutions to Bernstein Problems. In: Human Motor Action - Bernstein Reassessed (H.T.A. Whiting, Editor):373-412. Elsevier Science Publisher BV:Amsterdam. HOARE CAR (1978) Communicating Sequential Processes. Communications ACM 21(8):666- 667. HOFFER JA, and ANDREASSEN SS (1981) Regulation of Soleus Muscle Stiffness in Premammil- lary Cats: Intrinsic and Reflex Components. J. of Neurophysiology 45:267-265. HOGAN N (1980) Mechanical Impedance Control in Assistive Devices and Manipulators. Proceed. 1980 Joint Automatic Control Conf. (San Francisco). HOGAN N (1982) Programmable Impedance Control of Industrial Manipulators. Proceed. Conf. CAD/CAM in Mechanical Engineering (MIT, Cambridge, Massachusetts):186- 191. HOGAN N (1965) The Mechanics of Multi- Joint Posture and Movement Control. Biological Cybernetics 53:1-17. HOLLERBACH JM (1980) A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of

Dynamics Formulation Complexity. IEEE Trans. Systems Man Cybernetics SMC-11:730-736. HOLLERBACH JM, and FLASH T (1982) Dynamic Interaction between Limb Segments during Planar Arm Movements. Biological Cybernetics 44:67-77. HOLLERBACH JM, and ATKESON C (1985) Characterization of Joint Interpolated Arm Movements. Exp. Brain Res. HOLMES G (1930) The Cerebellum of Man. Brain 62:1-30. HOLTZMAN SR (1978) Music as System. Interface 7. HORAUD P, and BOLLES RC (1984) 3DPO's Strategy for Matching Three - Dimensional Objects in Range Data. Proceed. Intern. Conf. robotics (Atlanta, Georgia, Mar. 13-15):76~65. HORN BKP (1984) Extended Gaussian Images. Proceed. IEEE 72:1656-1878. HUGHES E (1983) Gymnastics for Giris. The Ronald Press:New York. HUTCHINSON GUEST A (1984) Dance Notation. Dance Books:London. HUXLEY HE (1963) Electron Microscope Studies on the Structure of Natural and Synthetic Protein Filaments from Striated Muscle. J. Molec. Biology 7:281 -308. HUXLEY HE (1969) The Mechanism of Muscular Contraction. Science 184:1356-1366. INMAN VT, and RALSTON HJ, TOOD F (1981) Human Walking. Williams and Wilkins:Baltimore. JACKSON JH (1884) Evolution and Dissolution of the Nervous System. In: Selected Writings of

 J.H. Jackson (J. Taylor, Editor, 1958):4S-7S. Basic Books: New York. JACKINS CL, and TANIMOTO SL (1980) Octrees and Their Use in Representing 3D Objects. Computer Graphics & Image Processing 14:349-270. JAIN R (1963) Dynamic Scene Analysis. In: Progress in Pattern Recognition (A. Rosenfeld and L. Kanal Editors) vol. 2. North Holland:Amsterdam. JEANNEROD M, and PRABLANC C (1983) Visual Control of Reaching Movements In Man. In: Motor Control Mechanisms in Health and Disease (J.E. Desmedt, Editor) - Advances in Neurology 39:13-30. Raven Press: New York. JENSEN CR, and SCHULTZ QW (1977) Applied Kinesiology. Mc Qraw Hill: New York. JERRY AJ (1977) The Shannon Sampling Theorem - its Various Extensions and Applications: A Tutorial Review. Proceed. IEEE, 65:1565-1595. JOHANNSON Q (1975) Visual Motion Perception. Scientific American 232:76-68. JOHNSON-LAIRD PN (1980) Mental Models in Cognitive Sciences. Cognitive Science 4:71- 115. JONES QM, and MILSUM JH (1965) Spatial and Dynamic Aspects of Visual Fixation. IEEE Trans. Biomedical Engineering BME-12:54 -62. JOYCE QC, RACK PMH, and WESTBURY DR (1969) The Mechanical Properties of Cat Soleus Muscle during Controlled Lengthening and Shortening Movements. J.

Physiology 204:461 - 474. KAO HSR, PINQ WAH L, NIANQ FENQ L, and SHEK DTL (1984) Chinese Calligraphy and Hear- trate Reduction. In: Psychological Studies of the Chinese Language (H.S.R. Kao and R. Hoosaln, Editors): 137-149. The Chinese Language Society: Hong Kong. KAPANDIJ IA (1970) The Physiology of the Joints. E&S Livingstone:London. KELSO JAS (1977) Motor Control Mechanisms underlying Human Movement Reproduction. J. Exper. Psychology - Human Perception and Performance 3: 529-543. KELSO JAS, SOUTHARD DL, and QOODMAN D (1979) On the Nature of Human Interlimb Coordination. Science 203: 1029-1031. KELSO JAS (1984) Phase Transition and Critical Behavior In Human Bimanual Coordination. American J. Physiology 246: 1000-104. KEPPEL E (1975) Approximating Complex Surfaces by Triangulation of Complex Lines. IBM J. Research & Development 19:2-11. KHATIB D (1981) Commande Dynamique dans I'Espace Operationelle des Robots Manipulateurs en Presence d'Obstacles. Docteur Ingenieur Thesis, Ecole National Superleur de I'Aeronautique e de l'Espace:Tolou8e. KINQSLEY EC, SCHOFIELD NA, and CASE K (1981) SAMMIE: A Computer Aid for Man - Machine Modeling. Computer Graphics 15: 163-169. KNUTTSON E, and RICHARDS C (1979) Different Types of Disturbed Motor Control in Gait of Hemiparetic Patients. Brain 102:405-430. KOREIN JU (1985) A Geometric Investigation of Reach. MIT Press: Cambridge Massachusetts.

KORNFELD WA, and HEWITT C (1981) The Scientific Community Metaphor. IEEE Trans. Systems Man Cybernetics SMC -11:339 -351. KUIPERS B (1984) Commonsense Reasoning about Causality: Deriving Behavior from Structure. Artificial Intelligence 24:169-204. KUPFERMAN I, and WEISS KR (1976) The Command Neuron Concept. The Behavioral and Brain Sciences 1:3-39. LABAN R (1956) Principles of Dance and Movement Notation. Macdonald & Evans:London. LACQUANITI F, and SOECHTING JF (1982) Coordination of Arm and Wrist Motion during a Reaching Task. J. Neuroscience 2:399 - 408. LAMOREUX LD, and SMITH LK (1962) Brunnstromm's Clinical Kinesiology. FA Davies Co:Philadelphia. LATHROP RL (1983) Parallelism in Manipulator Dynamics. MIT-AI Laboratory Technical Report 754, Cambridge, Massachusetts. LAWTON DT (1983) Processing Translational Motion Stereo. Computer Vision, Graphics and Image Processing 22:116-144. LE BOULCH J (1971) Vers une Science de Mouvement Humain: Introduction a la Psycho- cinetique. Les Edition

ESF.'Paris. LEE CSQ (1983) Robot Arm Kinematics, Dynamics, and Control. IEEE Computer 15: 62-81. LEE K (1984) Shape Optimization of Assemblies Using Geometric Properties. Ph.D. Thesis, MIT, Cambridge, Massachusetts. LIEBERMAN LI, and WESLEY MA (1977) AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly. IBM J. Research and Development 21:321 -333. LINDSAY PH, and NORMAN DA (1977) Human Information Processing. Academic Press: New York. LISSONI A, and BOCCARDI S (1982) Perspectives in Rehabilitation of the Adult Hemiplegic Patient. Rehabilitation and Learning 1:47-66. LIU D (1972) T'ai Chi Ch'uan and I Ching: A Choreography of Body and Mind. Routledge and Kegan PauhLondon. LOGAN BF (1977) Information in the Zero Crossing of Bandpass Signals. The Bell System TechnicalJ. 56:487-510. LOZANO PEREZ T (1982) Task Planning. In: Robot Motion: Planning and Control (Brady, Holler- bach, Johnson, and Lozano Perez, Editors). MIT Press:Cambridge, Massachusetts. LOZANO PEREZ T (1983) Robot Programming. IEEE Proceed. 71:821 -641. LUH JYS, WALKER MW, and PAUL RP (1980) On-line Computational Scheme for Mechanic Manipulators. Trans. ASME J. Dynamic Systems Measurement Control 120:69-76. MAC CONAIL MA, and BASMANJIAN JV (1989) Muscles and Movements: A Basis for Human Kinesiology. Williams & Wllkins: Baltimore.

 MAQNETAT-TALMAN N, and THALMAN D (1985) Computer Animation. Springer - Verlag:Tokyo. MAQOTT J (1984) Performance Evaluation of Concurrent Systems Using Petri Nets. Information Processing Letters 18:7-13. MAHONEY B (1984) Notation as a Tool for Video, Intern. Congress Movement Notation, Tel Aviv August 12-22. MALLQREN WR (1982) Formal Specification of Qraphic Data Types, ACM Trans. Programming Languages and Systems 4:687-710. MANDELBROT BB (1963) The Fractal Geometry of Nature. WH Freeman and Co:San Francisco. MARCH JG, SIMON HA (1963) Organizations. John Wiley & Sons:New York. MAREY EJ (1673) La Machine Animale, Locomotion Terrestre et Aerienne. Paris. MAREY EJ, and DEMENY G (1687) Etude Experimentale de la Locomotion Humaine. Comptes Rendus Hebdomadaires des Seances de I'Academie des Sciences 105: 544. MARINO G, MORASSO P, and ZACCARIA R (1985) Motor Knowledge Representation. Proceed. IJCAI-85 (Los Angeles, Califorbia, August 18-23):1110-1112. MARR D, POGGIO T, and ULLMAN S (1976) Bandpass Channels, Zero-Crossing, and Early Visual

Information Processing. MIT-AI Laboratory Memo 491, Cambridge, Massachusetts. MARR D, and POGGIO T (1979) A Computational Theory of Human Stereo Vision. Proc. Royal Society London B 204:301 -328. MARR D (1982) Vision. WH Freeman:San Francisco. MARSCHAK J, and RADNER R (1971) The Economic Theory of Teams. Yale Univ. Press. MARTIN WN, and AGGARWAL JK (1983) Volumetric Description of Objects from Multiple Views. IEEE Trans. Pattern Analysis Machine Intelligence PAMI -5:150-158. MASON MT (1981) Compliance and Force Control for Computer Controlled Manipulators. IEEE Trans. Systems Man Cybernetics SMC-11:416-432. MASSONE L, MORASSO P, and ZACCARIA R (1985) Shape from Occluding Contours. SPIE Proceed. 521:114-120. MASSONE L, SANDINI G, and TAGLIASCO V (1985) Form - Invariant Topological Mapping Strategy for 2D Shape Recognition. Computer Vision, Graphics and Image Processing 30:169-188. MAZER E (1983) LM-GEO: Geometric Programming of Assembly Robots. In: Advanced Software in Robotics (A. Danthine and M. Geradin, Editors). North - HollandiAmsterdam. MC KEON B, HOGAN N, and BIZZI E (1964) Effect of Temporary Path Constraint during Planar Arm Movements. Abstract 14th Annual Conf. Society for Neuroscience (Anaheim, California, Oct. 10-15):337. MEAGHER DJ (1961) Geometric Modeling using Octree Encoding. Computer Graphics Image Processing 19:129-147. MILANI COMPARETTI A, and GIDONI A (1971) Significato della Semeiotica Riflessologica per la Diagnosi Neuroevolutiva. Neuropsichiatria Infantile 121: 252-271.

 MILLER GA, and JOHNSON - LAIRD PN (1976) Language and Perception. Harvard Univ. Press:Cambridge Massachusetts. MINSKY M (1969) Perceptrons. MIT Press:Cambridge, Massachusetts. MINSKY M (1975) A Framework for Representing Knowledge. In: The Psychology of Computer Vision (P. Winston, Editor). McGraw Hill:NewYork. MOORE FR (1983) Introduction to Music Synthesis Using CMUSIC. Technical Report Univ. of San Diego:Calilornia. MORASSO P, BIZZI E, and DICHGANS J (1973) Adjustment of Saccade Characteristics during Eye Head Movements. Exper. Brain Res. 16: 548-563. MORASSO P (1981) Spatial Control of Arm Movements. Exper. Brain Res. 43:233-237. MORASSO P, and MUSSA IVALDI FA (1982a) Trajectory Formation and Handwriting: A Computational Model. Biological Cybernetics 45:131 -142. MORASSO P, and MUSSA IVALDI FA (1982b) Movement Planning for Robotic Manipulators. AICA

Annual Conf. (Padova Oct. 6-8). MORASSO P (1963) Three Dimensional Arm Trajectories. Biological Cybernetics 48:187-194. MORASSO P (1983) Coordination Aspects of Arm Trajectory Formation. Human Movement Science 2:197-210. MORASSO P, MUSSA IVALDI FA, and RUQQIERO C (1963) How a Discontinuous Mechanism Can Produce Continuous Movements. Acta Psychologies 54:83-96. MORASSO P, and TAGLIASCO V (1983) Images in Robotics Smooth Pursuit. SPIE Proceed. 449:29-35. MORASSO P, TROIANO E, and ZACCARIA R (1985) Representing Motor Knowledge. In: Artificial Intelligence - Methodology, Systems, Applications (W.Bibel, B. Petkoff, Editors). North- Holland:Amsterdam. MORASSO P (1985) Solid Modeling in Robotic Vision. IFIP Working Conf. Robots in Discrete Manufacturing (Somalvico, Qini, and Mc Waters, Editors). North Holland:Amsterdam. MORASSO P, and SANDINI Q (1985) 3D Reconstruction from Multiple Stereo Views. Ill Intern. Conf. Image Analysis and Processing (Rapallo, October 1 - 2). MORTON DJ (1952) Human Locomotion and Body Form: A Study of Gravity and Man. Williams & Wllkins:Baltimore. MUKAROVSKY J (1944) The Essence of the Visual Arts. In: Semiotics of Art (L Matejka and I.R. Titunik, Editors, 1976): 229-244. MIT Press:Cambridge, Massachusetts. MURRAY MP (1967) Gait as a Total Pattern of Movement. American J. Physical Medicine. 46:290. MUSSA IVALDI FA, HOGAN N, and BIZZI E (1984) Invariant Features of Hand Postural Stiffness. Abstract 14th Annual Conf. Society for Neuroscience (Anaheim, California, October 10- 15):l02.3,p.337. MUSSA IVALDI FA, BIZZI E, and HOGAN N (1985) Neural, Mechanical, and Geometric Factors

 Subserving Arm Postures in Humans. J. Neuroscience 5:2737-2743. MUYBRIOQE E (1901) The Human Figure in Motion. Chapman & Hall:London. MYERS QJ (1982) Advances in Computers. John Wiley & Sons:NewYork. NACKMAN RL (1984) Two - Dimensional Critical Point Configuration Graphs. IEEE Trans. Pattern Analysis Machine Intelligence PAMI - 6:442 - 449. NASHNER LM, and QRIMM RJ (1978) Analysis of Multiloop Dyscontrols in Standing Cerebellar Patients. In: Cerebral Motor Control in Man - Long Loop Mechanisms (J.E. Desmedt, Editor). Progress in Neurophysiology 4: 300-319. KargerBasel. NASHNER LM (1980) Balance Adjustment of Humans Perturbed while Walking. J. Neurophysiology 44:650 - 684. NASHNER LM, and MC COLLUM (1985) The Organization of Human Postural Movements. The Behavioral and Brain Sciences. 8: 559 - 560.

NEUMAN B (1980) Exploiting Image Formation Knowledge for Motion Analysis. IEEE Trans. Pattern Analysis Machine Intelligence PAMI-2:550-554. NEVATIA R, and BINFORD TO (1977) Description and Recognition of Curved Objects. Artificial Intelligence 8:77-98. NEWELL A, and SIMON HA (1972) Human Problem Solving. Prentice Hall:Englewood Cliff, New Jersey. NICHOLS TR, and HOUK JC (1976) Improvement in Linearity and Regulation of Stiffness that Results from Actions of Stretch Reflex. J. Neurophysiology 39:119-142. NIELSEN M, PLOTKIN Q, and WINSKEL Q (1981) Petri Nets, Event Structures and Domains. Part I. Theoretical Computer Science 13:85-108. NILSSON NJ (1980) Principles of Artificial Intelligence. Tioga Publishing Co:Palo Alto. NISHIHARA HK (1983) PRISM: A Practical Realtime Imaging Stereo Matcher. Proceed. IFS- SPIE Intern. Conf. Robot Vision Sensory Control (Cambridge Massachusetts November 6- 10):121-130. NOE JD, and NUTT QJ (1973) Macro-E-Nets for Representations of Parallel Systems. IEEE Trans. Computers C-22:8. ORIN DE, MCQHEE RB, VUKOBRATOVICH M, and HARTOCH Q (1979) Kinematic and Kinetic Analysis of Open-Chain Linkages utilizing Newton-Euler Methods. Mathematical Biosciences 43:107-130. O'ROURKE J, and BADLER N (1979) Decomposition of Three - Dimensional Objects into Spheres. IEEE Trans. Pattern Analysis Machine Intelligence PAMI -1:295 -305. O'ROURKE O (1981) Triangulation of Minima Area as 3D Object Models. IJCAI-81 (Vancouver):664 - 666. PALMIERI Q, OLIVA QA, and SCOTTO M (1971) CRT Spot Follower Device for Eye Movement Measurement. Kybemetik 8; 23-30. PARNAS (1984). ACM Forum. Communications of the ACM 27:1155-1158.

References 371 PAUL RP (1981) Robot Manipulators: Mathematics, Programming and Control. MIT Press: Cambridge, Massachusetts. PAUL RP, and SHIMANO B (1976) Compliance and Control. Proceed. Joint Automatic Control Conf. (San Francisco): 694-699. PAVLiDIS T (1983) Algorithms for Graphics and Imege Processing. Springer Verlag:Berlin. PAYNTER HM (1961) Analysis and Design of Engineering Systems. MIT Press:Cambridge. PETERSON JL (1981) Petri Net Theory and the Modeling of Systems. Prentice Hall:Englewood Cliffs, New Jersey. PETRI CA (1963a) Communication mit Automaten. Schriften des Institutes fur Instrumentelle Mathematik, Bonn. PETRI CA (1962b) Fundamentals of a Theory of Asynchronous Information Flow and Information Processing. Proceed. IFIP

Congress 63 (Munich):386-390. North Holland:Amsterdam. PETRI CA (1976) General Net Theory. Proceed. Joint IBM & Univ. of Newcastle upon Tyne Seminar on Computer Systems Design (Sept. 1976). PETRI CA (1977) Non Sequential Processes. Technical Report GMD - ISF - 77 - 5, GMD, Bonn. PETRI CA (1978) Concurrency as a Basis of System Thinking. Technical Report ISF -78 -8, GMD, Bonn. PIAGET J (1946) Les Notion de Mouvement et de Vitesse chez I'Enfant. Presses Universitaire de France:Paris. PIAGET J (1971) Biology and Knowledge: An Essay on the Relation between Organic Regulations and Cognitive Processes. Edinburgh Univ. Press:Edinburgh. PIPITONE FJ, and MARSHALL TG (1983) A Wide-Field Scanning Triangulation Rangefinder for Machine Vision. Intern. J. Robotics Res. 3:39-49. PLESS E, and PLUNNECKE H (1980) A Bibliography of Net Theory. Second Edition ISF-Report 80.05, GMD, Bonn. POGGIO T, and TORRE V (1984) III-Posed Problems and Regularlzatlon Analysis in Early Vision. MIT - Al Laboratory Memo 773, Cambridge, Massachusetts. POIZNER H, NEWKIRK D, and BELLUGI U (1983) Processes Controlling Human Movements: Neuromotor Constraints on American Sign Language. J. Motor Behavior 15: 2-18. POLIT A, and BIZZi E (1978) Processes Controlling Arm Movements in Monkeys. Science 201:1235-1237. POLIT A, and MASSION J (1979) Patterns of Postural Support during Limb Movement. Soc. Neuroscience 5: 383. POLIT A, and BIZZI E (1982) Characteristics of Motor Programs underlying Arm Movements in Monkeys. J. Neurophysiology 42:183-194. POPPLESTONE Rj, AMBLER Ap, and BELLOS I (1978) RAPT: A Language for Describing Assemblies. Industr. Robot 5:131-137.

 POPPLESTONE RJ, AMBLER AP, and BELLOS I (1982) An Interpreter for a Language for Describing Assemblies. In: Robot Motion - Planning and Control (M. Brady, J. Holler bach, T. Johnson, T. Lozano Perez, M.T. Mason, Editors):517-566. MIT Press:Cambridge, Massachusetts. PORKERT M (1974) The Theoretical Foundation of Chinese Medicine. MIT Press:Cambridge, Massachusetts. POTTER TE, and WILLMERT KD (1975) Three - Dimensional Human Display Model. Computer Graphics 9:102-110. PRIQOQINE I (1973) Time, Irreversibility, and Structure. In: The Physicist's Conception of Nature (J. Mehra, Editor):561-593. Dordrecht-Holland:Boston, Massachusetts. PYLYSHYN ZW (1984) Computation and Cognition. MIT Press: Cambridge. RAQQHIANTI CL (1946) Commenti di Critica d'Arte. Boringhieri:Torino. RACK PMH, and WESTBURY DR (1969) The Effects of

Length and Stimulus Rate on Tension in the Isometric Cat Soleus Muscle. J. Physiology 204:443-460. RAIBERT MH, and CRAIQ JJ (1981) Hybrid Position/Force Control Manipulators. Trans. ASME J. Dynamic Systems Measurement Control 102:126-133. RAIBERT MH, and HORN BKP (1978) Manipulator Control using the Configuration Space Method. Industrial Robot 5:69 - 73. RAMAMOORTHY CV, and HO GS (1980) Performance Evaluation of Asynchronous Concurrent Systems using Petri Nets. IEEE Trans. Software Engineering SE-6, 5:440-449. RAMCHANDANI C (1974) Analysis of Asynchronous Concurrent Systems by Petri Nets. MIT Project MAC Technical Report 120, Cambridge, Massachusetts. REED (1984) From Action Gestalts to Direct Action. In: Human motor action - Bernstein reassessed (H.T.A. Whiting, Editor): 157-170. Elsevier Science Publlsher:Amsterdam. REISIG W (1982) Petrinetze. Springer Verlag:Berlin. REKTORIS K (Ed.) (1969) Survey of Applicable Mathematics. MIT Press:Cambridge Massachusetts. REQUICHA AG (1980) Representation of Rigid Solids: Theory, Methods and Systems. ACM Computing Surveys 12:437-475. REYNOLDS CW (1962) Computer Animation with Scripts and Actors. Proc. SIGGRAPH-82 Computer Graphics 16:289-296. RIEMANN H (1884) Musikalische Dynamic und Agogik. Amburg - Pietroburg. ROBERTS TD (1976) The Role of Vestibular and Neck Receptors In Locomotion. In: Neural Control of Locomotion (R. Herman, S. Grillner, P. Stein, D. Stuart, Editors). Plenum Press: New York. ROBINSON DA (1984) The Mechanics of Human Saccadic Eye Movements. J. Physiology 174:245-264. ROBINSON DA (1982) The Uee of Matrices in Analyzing the 3D Behavior of the Vestlbulo Ocular Reflex. Biological Cybernetics 48: 55-66.

 RODET X, and COINTE P (1984) FORMES: Composition and Scheduling of Processes. Computer Music J. 8:32 - 50. ROSA LA (1929) Espressione Mimica. Manuali HceplhMilano. ROSEMBERQ RC, and KARNOPP DC (1983) Introduction to Physical System Dynamics. McGraw HilliNew York. SACERDOTI ED (1977) A Structure for Plans and Behavior. Elsevier Publlsher:New York. SALISBURY JK (1980) Active Stiffness Control of a Manipulator in Cartesian Coordinates. IEEE Conf. Decision and Control (Albuquerque). SALISBURY JK (1982) Kinematics and Force Analysis of Articulated Hands. Ph.D. Thesis, Stanford Univ., Stanford, California. SALTZMAN E (1979) Levels of Sensorimotor Representation. J. Mathematical Psychology 20: 91-163. SALTZMAN E, and KELSO JAS

(1983) Skilled Actions: A Task Dynamic Approach. Hasking Laboratory, Status Report on Speech Research. SALTZMAN E (1985) Task Dynamic Coordination of the Speech Articulators: A Preliminary Model. Exp. Brain Res. Supplementum. SCHLEMMER O (1971) Man. MIT Press:Cambridge Massachusetts. SCHMID AB (1976) Modern Rhythmic Gymnastics. Mayfield Publishing Co: Palo Alto California. SCHMIDT RA (1975) A Schema Theory of Discrete Motor Skill Learning. Psychological Review 82:225-260. SCHUDY RB, and BALLARD DH (1978) Model Detection of Cardiac Chambers in Ultrasound Images. Computer Science Dept., Univ. Rochester, Techical Report 12. SERRA J (1982) Mathematical Morphology. Academic Press:New York. SHEPARD RN, and COOPER LA (1982) Mental Images and Their Transformations. The MIT Press:Cambridge Massachusetts. SIKLOSSY L, and DREUSSI J (1973) An Efficient Robot Planner which Generates its own Procedures. Proceed. Third IJCAl:423-430. SILVER WM (1982) On the Equivalence of the Lagrangian and Newton-Euler Dynamics of Manipulators. Intern. J. Robotics Res. 1:60-70. SMOLIAR SW (1971) A Parallel Processing Model of Musical Structures. Ph.D. Thesis, MIT, Cambridge, Massachusetts. SOECHTING JF, and LACQUANITI F (1981) Invariant Characteristics of a Pointing Movement in Man. J. Neuroscience 1:710-720. SRIHARI SN (1981) Representation of Three - Dimensional Digital Images. ACM Computing Surveys 13:399-424. STEFIK MJ (1981) Planning with Constraints. Artificial Intelligence 2:111-139.

 STELMACH QE, and HUGHES BG (1984) Cognitivism and Future Theories of Action. In: Cognition and Motor Processes (W. Prinz & A.F. Sanders, Editors) Springer Verlag:Berlin. STETSON RH, and Mc DILL JA (1923) Mechanisms of Different Types of Movements. Psychol. Monograph 32: 18. SUSSMAN GJ, and MC DERMOTT DV (1972) From PLANNER to CONNIVER - A Genetic Approach. Proceed. Fall Joint Computer Conf., AFIPS Press:1171-1179. SUSSMAN GJ (1975) A Computer Model of Skill Acquisition. American Elsevier:New York. SVETKOFF DJ, LEONARD PF, SAMPSON RE, and JAIN RC (1984) Techniques for Real-Time 3D Feature Extraction using Range Information. Proceed. SPIE Symposium Intelligent Robotics and Computer Vision (Cambridge, Massachusetts):512. TARDIEAU G (1973) The Equinus in Cerebral Palsy. English edn. Cahiers du C.D.I.:1,S. TARDIEAU G (1973) Factorial Analysis as an Approach to Cerebral Palsy. English edn. Cahlers du C.D.I.:2. TEULINGS HL, and THOMASSEN JWM

(1979) Computer Aided Analysis of Handwriting Movements. Visible Language 13:218-231. TEULINGS HL, and MAARSE FJ (1984) Digital Recording and Processing of Handwriting Movements. Human Movement Science 3: 193-217. THOMPSON JA (1980) How Do We Use Visual Information to Control Locomotion. Trends in Neuroscience 3:247-250. TILOVE RB (1980) Set Membership Classification: A Unified Approach to Geometric Intersection Problems. IEEE Trans. Computer C- 29:674 -883. TANGWONGSAN S, and FU KS (1979) An Application of Learning to Robot Planning. Intern. J. Computer & information Sciences 8:303 - 333. TRELEAVEN PC, BROWNBRIDGE DR, and HOPKINS RP (1982) Date-Driven and Demand - Driven Computer Architecture. ACM Computing Surveys 14:93-145. TURVEY MT, and KUGLER PN (1984) An Ecological Approach to Action and Perception. In: Human Motor Action - Bernstein Reassessed (H.T.A. Whiting, Editor): 373-412. Elsevier Science Publisher BV:Amsterdam. UDUPA JK (1983) Display of 3D Information in Discrete 3D Scenes Produced by Computerized Tomography. Proceed, of the IEEE 71. UDUPA SM (1977) Collision Detection and Avoidance in Computer Controlled Manipulators. IJCAI-77 (Cambridge, Massachusetts). ULLMAN (1980) Against Direct Perception. The Behavioral and Brain Sciences 3: 373-415. VIVIANI P, and TERZUOLO CA (1980) Space-Time Imparlance in Learned Motor Skills. In: Tutorials in Motor Behavior (G.E. Stelmach, J. Requln, Editors): 525-533. North Holland:Amsterdam. VIVIANI P, and TERZUOLO C (1982) Trajectory Determines Movement Dynamics. Neuroscience 7:431-437.

 VON NEUMANN J (1958) The Computer and the Brain. Yale Univ. Press:New Haven, Connecticut. WATSON LT, LAFFEY TJ, and HARALICK RM (1985) Topographic Classification of Digital Image Intensity Surfaces using Generalized Splines and the Discrete Cosine Transformation. Computer Vision Graphics and Image Processing 29:143-167. WATSON PC (1976) A Multidimensional System Analysis of the Assembly Process as performed by a Manipulator. C.S. Draper Laboratory Report P-3B4, Cambridge, Massachusetts. WHITNEY DE (1972) The Mathematics of Coordinated Control of Prostheses and Manipulators. Trans. ASME J. Dynamic Systems Measurement Control 94:303-309. WHITNEY DE (1982) Quasi - Static Assembly of Compliantly Supported Rigid Parts. Trans. ASME J. Dynamic Systems Measurement Control 104:65-77. WIENER N. (1947)

Cybernetics. MIT Press:Cambridge, Massachusetts. WICKSTROM RL (1970) Fundamental Motor Patterns. Lea & Febiger:Philadelphla. WILLIAMS M, and LISSNER HR (1962) Biomechanics of Human Motion. WB Saunders:Philadelphia.

This Page Intentionally Left Blank

 SUBJECT INDEX

ADVANCES IN PSYCHOLOGY 33 •V" t-r P Morasso V.Tagliasco Editors m u t ■ • 1 North Holland