Matching Tensors for Automatic Correspondence and Registration

Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens

School of Computer Science and Software Engineering
The University of Western Australia, Crawley, WA 6009, Australia,
{ajmal, bennamou, robyn}@csse.uwa.edu.au,
http://www.cs.uwa.edu.au

Abstract. Complete 3-D modeling of a free-form object requires acquisition from multiple view-points. These views are then required to be registered in a common coordinate system by establishing correspondence between them in their regions of overlap. In this paper, we present an automatic correspondence technique for pair-wise registration of different views of a free-form object. The technique is based upon a novel robust representation scheme reported in this paper. Our representation scheme defines local 3-D grids over the object's surface and represents the surface inside each grid by a fourth order tensor. Multiple tensors are built for the views which are then matched, using a correlation and verification technique to establish correspondence between a model and a scene tensor. This correspondence is then used to derive a rigid transformation that aligns the two views. The transformation is verified and refined using a variant of ICP. Our correspondence technique is fully automatic and does not assume any knowledge of the viewpoints or regions of overlap of the data sets. Our results show that our technique is accurate, robust, efficient and independent of the resolution of the views.

1 Introduction

Three dimensional modeling of objects has become a requirement in a large number of fields ranging from the entertainment industry to medical science. Various methods are available for scanning views of 3-D objects to obtain 2.5-D images in the form of a cloud of points (see Fig. 1), but none of these methods can completely model a free-form object with a single view due to self occlusion. Multiple overlapping views of the object must be acquired to complete the 3-D model. These views are then required to be registered in a common coordinate system, but before they can be registered, correspondence must be established between the views in their regions of overlap. Points on two different views that correspond to the same point on the object are said to be corresponding points. These correspondences are then used to derive an optimal transformation that aligns the views. The automatic correspondence problem is difficult to tackle due to two main reasons. First, there is no knowledge of the viewing angles and second, there is no knowledge about the regions of overlap of the views. The latter implies that every point on one view does not necessarily have a

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 495–505, 2004.

corresponding point in the other view and that there is no a priori knowledge of correspondences.

Existing techniques of correspondence are based on various assumptions and are not fully automatic [1]. The classic Iterated Closest Point (ICP) algorithm [2], Chen and Medioni's algorithm [3] and registration based on maximizing mutual information [4] all require initial estimates. In case the initial estimate is not accurate, these techniques may not converge to the correct solution. Some techniques like the RANSAC-based DARCES [5] are based upon exhaustive search and are not efficient. Bitangent curve matching [6] calculates first order derivatives which are sensitive to noise and require the underlying surface to be smooth. Moreover, bitangent curves are global features and may not be fully contained inside the overlapping region of the views. Three tuple matching [7] calculates the first and second order derivatives which are sensitive to noise and require the underlying surfaces to be smooth. SAI matching [8] requires the underlying surfaces to be free of topological holes. Geometric histogram matching [9] makes use of a 3-D Hough transform [10] which is computationally expensive. Roth's technique [11] relies upon the presence of a significant amount of texture on the surface of the object for consistent extraction of feature points from their intensity images. Matching oriented points [12] uses spin image representation which is not unique and gives a lot of ambiguous correspondences. These correspondences must be processed through a number of filtration stages to prune out incorrect correspondences making the technique inefficient.

In this paper, we present a fully automatic correspondence technique which does not assume any prior knowledge of the view-points or the regions of overlap of the different views of an object. It is applicable to free-form objects and does not make assumptions about the shape of the underlying surface. Our technique is inspired by the spin image representation [13]. However, instead of making 2-D histograms of vertex positions, we represent the surface of the object in local 3-D grids. This results in a unique representation that facilitates accurate correspondences. The strength of our technique lies in the new representation scheme that we have developed. Our correspondence technique starts by converting two views of an object, acquired through a 3-D data acquisition system, into triangular meshes. Normals are then calculated for each point and triangular facet. Sets of two points along with their normals on each triangular mesh are then selected to define 3-D grids over the surface. The surface area and normal information in all the bins of each grid is then stored in a tensor. These tensors are matched to establish correspondences between the two views. Tensors that give the best match are then used to compute a rigid transformation that aligns the two views. This transformation is refined using a variant of the ICP algorithm [15].

The rest of this paper is organized as follows. In Section 2 we describe our new 3-D free-form object representation scheme. In Section 3 we explain the matching process to establish correct correspondences between the two views. Section 4 gives details of our experimental results. In Section 5 we discuss and analyze our results. Finally conclusions are given in Section 6.

2 A New Representation Scheme Based on Tensors

In this section we will describe our new tensor based 3-D free-form object representation scheme. Before we construct the tensors, the n data points are first converted into triangular meshes and normals are calculated at each vertex and triangular facet. This information is stored in a data structure along with the neighbourhood polygons information for each point and each polygon. Next a set of two points, along with their normals are selected to define a 3-D coordinate basis. To avoid the C_2^n combinatorial explosion of the points, we select points that are at a certain fixed distance from each other. This distance is defined as a multiple of the mesh resolution. In our experiments we have set this distance to four times the mesh resolution, which is far enough to make the calculation of the coordinate basis less sensitive to noise and close enough for both points to lie inside the region of overlap. To speed up the search for such points we consider points that are four edges away from each other. This can be easily performed by checking the fourth and fifth neighbourhood of the point under consideration. The center of the line joining the two points defines the origin of the new 3-D basis. The average of the two vectors defines the z-axis, since we want the z axis to be pointing away from the surface. The cross product of the two vectors defines the x-axis and finally the cross product of the z-axis with the x-axis defines the y-axis.

This 3-D basis and its origin is used to define a 3-D grid centered at the origin. Two parameters need to be selected, namely, the number of bins in the 3-D grid and the size of each bin. Varying the number of bins from less to more varies the representation from being local to global. We have selected the number of bins to be 10 in all the directions making the grid take the shape of a cube. The bin size defines the level of granularity at which the information about the object's surface is stored. The bin size is kept as a multiple of the mesh resolution because the mesh resolution is generally related to the size of the features on the object. In our experiments we have selected a bin size equal to the mesh resolution.

Once the 3-D grid is defined (see Fig. 1 and 2) the area of the triangular mesh intersecting each bin is calculated along with the average surface normal of the surface at that position. Next the angle between this surface normal and the z-axis of the grid is calculated. This angle is an estimate of the curvature of the surface at that point. This area and angle information is stored in a fourth order tensor which corresponds to a local representation of the surface in the 3-D cubic grid. To find the area of intersection of the surface with each cubic bin, we start from one of the two points that were used to define the 3-D grid and visit each triangle in its immediate neighbourhood. Since the points are approximately two mesh resolutions away from the origin they are bound to be inside the 3-D grid. The area of intersection of the triangle and a cubic bin of the grid is calculated using Sutherland Hodgman's algorithm [14]. Once all the triangles in the immediate neighbourhood of the point have been visited and their intersection with the grid bins has been calculated, the neighbourhood triangles of these triangles are visited. This process continues until a stage is reached when all the neighbouring triangles are outside the 3-D grid at which point the

computation is stopped. While calculating the area of intersection of a triangle with a cubic bin the angle between its normal and the z-axis is also calculated and stored in the fourth order tensor. Since more than one triangle can intersect a bin, the calculated area of intersection is added to the area already present in that bin, as a result of its intersection with another triangle. The angles of the triangular facets, crossing a particular bin, with the z-axis are averaged by weighting them by their corresponding intersection area with that bin.

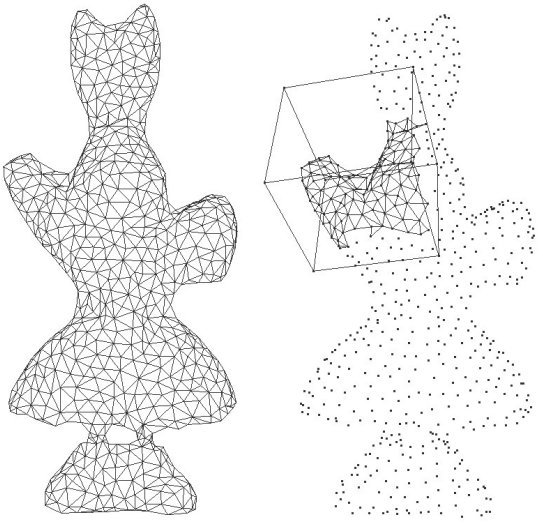

Fig. 1. Left: Data points of a view of the bunny converted into a triangular mesh. (Data courtesy of the Robotics Institute, CMU) Right: The cube represents the boundary of a 3-D grid. Only the triangles that contribute toward the tensor corresponding to this grid, as shown, are considered in the calculation of the tensor corresponding to this grid.

3 Matching Tensors for Correspondence and Registration

Since a tensor corresponds to a representation of the 3-D surface inside an object centered grid, different views of the same surface will have similar tensors. Minor differences may exist between these tensors as a result of different possible triangulations of the same surface due to noise and variations in sampling. However, corresponding tensors will have a better match as compared to the non-corresponding tensors. We use the linear correlation coefficient to match tensors. Corresponding tensors will give a high correlation coefficient and can easily be differentiated.

To establish correspondence between a model view and a scene view of an object, first the tensors for all the point pairs of the model (that are four times

the mesh resolution apart) are calculated. Restricting the selection of point pairs significantly reduces the number of possible pairs from C_2^n. The search for such points is speeded up by searching the fourth and fifth neighbourhood of the points only. Next a point is selected at random from the scene and all possible points that can be paired with it are identified. A tensor is then calculated for the first point and one of its peers. This tensor is then matched with all the tensors of the model. If a significant match is found, the algorithm proceeds to the next stage of verification, else it drops this tensor and calculates another tensor using the first point and another one of its remaining peers. The tensors are matched only in those bins where both tensors have surface data (this approach has also been used by Johnson [12]). This is done to cater for situations where some part of the object may be occluded in one view. Matching proceeds as follows. First, the overlap ratio R_O of the two tensors is calculated according to Equation 1. If R_O is greater than a threshold t_r, the algorithm proceeds to calculate the correlation coefficient of the two tensors in their region of overlap. If R_O is less than t_r, the model tensor is not considered for further matching. In our experiments we found that $t_r = 0.6$ gave good results.

$$R_O = \frac{\sum I_{sm}}{\sum U_{sm}} \tag{1}$$

In this Equation I_{sm} is the intersection of the occupied bins of the scene and the model tensor. U_{sm} is the union of the occupied bins of the scene and the model tensor.

If the correlation coefficient of the scene tensor with some model tensors is significantly higher than the correlation coefficients with the remaining model tensors, then all such model tensors are considered to be potential correspondences. Such model tensors are taken to be those having correlation coefficient two standard deviations higher than the mean correlation coefficient of the scene tensor with all the model tensors. The best matching tensor is verified first. Verification is performed by transforming one of the two scene points, used to calculate the tensor, to the model coordinate system. This transformation is calculated by transforming the corresponding 3-D basis of the scene tensor to the 3-D basis of the model tensor (Eqn. 2 and Eqn. 3).

$$\mathbf{R} = \mathbf{B_s^T B_m} \tag{2}$$

$$\mathbf{t} = \mathbf{O_m} - \mathbf{O_s R} \tag{3}$$

$\mathbf{B_m}$ and $\mathbf{B_s}$ are the matrices of coordinate basis used to define the model and scene tensors respectively. $\mathbf{O_m}$ and $\mathbf{O_s}$ are the coordinates of the origins of the model and scene grids in the coordinate basis of the entire scene and model respectively. $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector that will align the scene data with the model data.

Next the distance between the transformed scene point and its corresponding point in the model tensor is calculated. If this distance is below a threshold d_{t1}, i.e. the scene point is close to its corresponding model point, the verification process proceeds to the next step, else the model tensor is dropped and the next

model tensor with the highest correlation coefficient is tested. Figure 2 shows an incorrect transformation calculated as a result of matching tensors. The two points that were used to calculate the tensors are connected by a line. These points are not close to their counter-parts in the other tensor, representing a poor tensor match. Figure 2 also shows a correct transformation calculated as a result of matching tensors. Here the two points are very close to their counter-parts in the other tensor representing a good tensor match. In our experiments we set d_{t1} equal to one fourth of the mesh resolution to ensure that only the best matching tensors pass this test. If all model tensors fail this test, another set of two points is selected from the scene and the whole process is repeated.

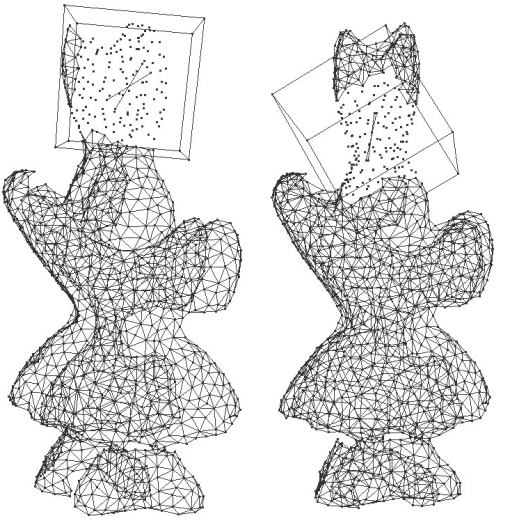

Fig. 2. Two views of the bunny registered using a single set of matching tensors. The bounding box is the region where the scene and model tensors are matched. The points used to calculate the 3-D basis are joined by a line. Left: These points are not close to their counter-parts resulting in an inaccurate transformation. Right: The points are close to their counter-parts in the other view, resulting in a good tensor match and an accurate transformation.

In case the distance between the two pairs of points is less than d_{t1}, all the scene points are transformed to the model coordinate system using the same transformation. The transformation resulting from a single set of good matching tensors is accurate enough to establish scene point to model point correspondences on the basis of nearest neighbour. The search for nearest neighbour starts from the scene points that are connected directly to one of the initial points used to define the 3-D basis. Scene points that have a model point within a distance of d_{t2} are turned into correspondences. We chose d_{t2} equal to the mesh resolution in our experiments. d_{t2} is selected considerably higher than d_{t1} since the initial transformation has been calculated based on a single set of matching tensors.

Even a small amount of error in this transformation will cause greater misalignment between the scene and the model points that are far from the origin of the tensors. Next correspondences are found for more scene points that are directly connected to the points for which correspondences have recently been found. This process continues until correspondences are spread throughout the mesh and no more correspondences can be found. If the total number of correspondences at the end is more than half the total number of scene points, the initial transformation given in Equations 2 and 3 is accepted and refined by applying another transformation calculated from the entire set of correspondences found during the verification process.

4 Experimental Results

We have performed our experiments on a large data set. The results of only four objects are reported in this paper. The data set (in the form of a cloud of points) of the first three of these objects namely, the bunny, the truck and the robot was provided by the Robotics Institute, Carnegie Mellon University, whereas the data of Donald Duck was acquired using the Faro Arm acquisition system in our laboratory. The gray scale pictures of these objects are shown in Figure 3. Three views of the first three objects were taken and our automatic correspondence algorithm was applied to register these views. Figure 4 shows the results of our experiments. Each row of Figure 4 contains a different object. The first three columns of Figure 4 contain the three different views of these objects and the fourth column contains all three views registered in a common coordinate frame. The registered views are shown in different shades so that they can be differentiated after alignment.

Fig. 3. Gray scale pictures of the bunny, the truck, the robot and Donald Duck used in our experiments.

We have also tested our algorithm on data sets where each view is acquired at different resolutions and in each case it resulted in an accurate registration. This shows that our algorithm is independent of the resolution of the data. Unlike spin image matching our correspondence algorithm does not require uniform mesh resolution and is therefore robust to variations in resolution within a single view. Figure 5 shows the result of our algorithm on the Donald Duck data set

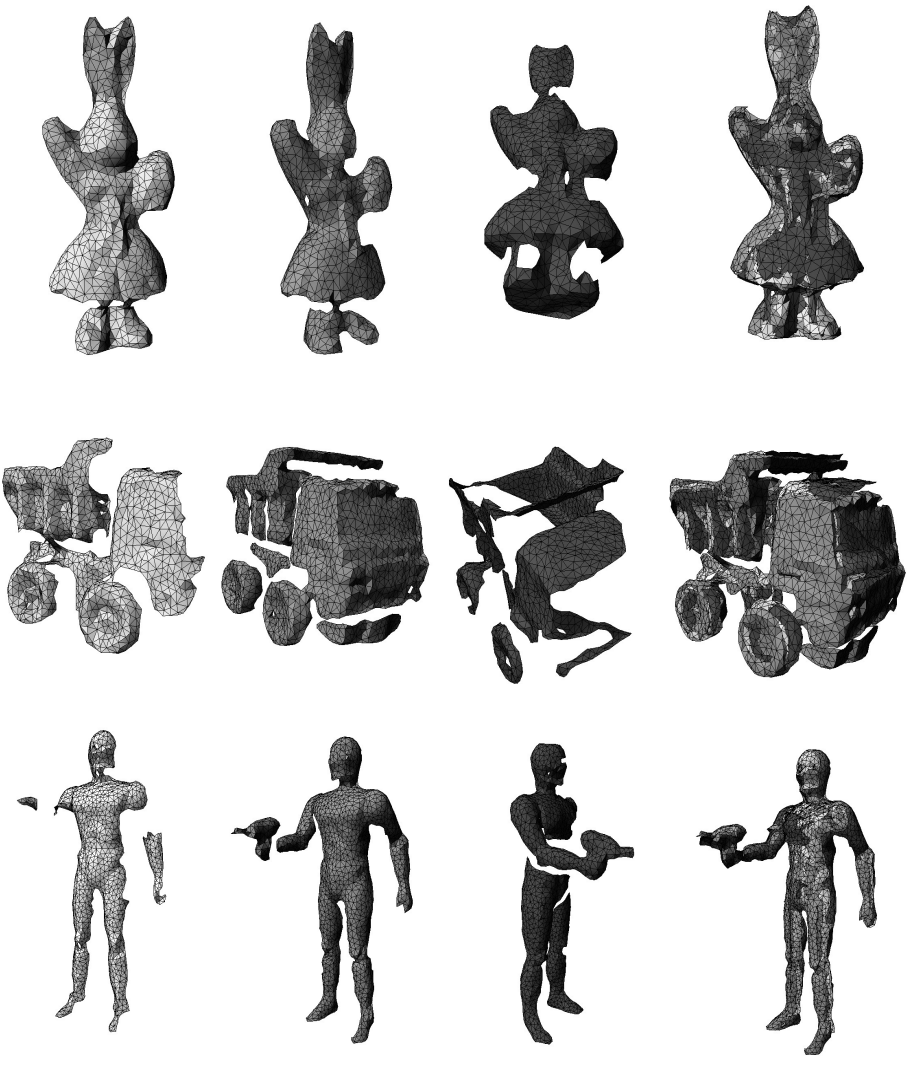

Fig. 4. Registration results. Each row contains a separate object, namely the bunny, the truck and the robot. The first three columns of the Figure represent three views of the respective objects in different grey shading. The last column shows the registered views. Notice the contribution of each view shown in different shadings in the registered view.

that has an extremely non-uniform mesh resolution with edge lengths varying from a minimum of $0.2mm$ to a maximum of $12.8mm$ and a standard deviation of $2.4mm$. The registration result in this case is an indication of the extent of robustness of our algorithm to variations in the resolution of data set. Such variations are commonly expected from all sensors when there are large variations

in the orientation of the surface. Data points of a surface patch that is oblique to the sensor will have low density as compared to a surface patch that is vertical to the sensor.

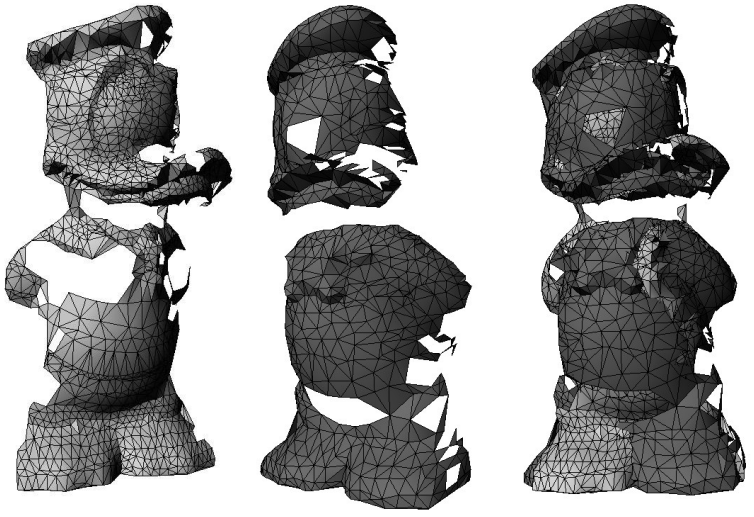

Fig. 5. Result of the algorithm applied to two views of Donald Duck having non-uniform mesh resolution and patches of missing data. Accurate registration is obtained in the presence of missing data. The missing data is due to the difficulty in scanning with a contact sensor (the Faro Arm in our case).

5 Discussion and Analysis

Our tensor based representation scheme is robust to noise due to the following reasons. The 3-D basis is defined from two points that are four mesh resolutions apart. The origin is defined by the center of the line joining the two points which reduces the effect of noise. The z-axis is defined as the average of the normals of these points, hence reducing the effect of noise and variations in surface sampling. Quantization is performed by dividing the surface area into the bins of a 3-D grid. This significantly reduces the effect of different possible triangulations of the surface data. The use of a statistical matching tool (in our case the correlation coefficient) performs better in the presence of noise as compared to linear matching techniques. All these factors ensure that the tensors representing the same surface in two different views will give high similarity as compared to tensors representing different surface regions.

We have taken the following measures to ensure that our algorithm is efficient in terms of memory consumption and performance. First, to avoid the combinatorial explosion of pairing points, we only consider points that are four mesh

resolutions apart with some tolerance. This restricts the possible pairs of points to $O(n)$ instead of $O(n^2)$. Next, to calculate the area of the mesh inside the individual bins of the 3-D grid, instead of visiting each bin of the grid, we start from one of the points used to define it and consider its neighbouring polygons that are inside the grid. These polygons are visited one at a time and their area of intersection with the bins is calculated using an efficient algorithm (Hodgman [14]).

During the matching phase, in order to speed up the process, two tensors are only matched if their overlap ratio R_O is more than 0.6. If a match with a high correlation coefficient is found, it is verified by transforming only one of the scene points, used to define the 3-D basis. If the distance of the transformed point is less than d_{t1} from its corresponding model point, the transformation is accepted, otherwise it is rejected. This verification step is very fast since the rotation matrix and the translation vector can easily be calculated from Equations 2 and 3. Choosing d_{t1} equal to 1/4th of the mesh resolution ensures that only a good match passes this test. This verification step almost always identifies an incorrect tensor match and saves the algorithm from proceeding to the verification stage. Verification of the transformation involves the search for the nearest neighbour of every scene point, which is computationally expensive. Our algorithm does not find a list of correspondences and pass them through a series of filtration steps as in the case of the spin images approach [12]. Instead it selects a scene tensor and finds its matching tensor in the model. If the match passes the above explained verification step, it proceeds to verify the transformation derived from Equations 2 and 3. This transformation is refined and the algorithm stops. The algorithm does not have to visit every possible correspondence and is therefore less computationally expensive.

Once the views are registered they can easily be integrated and reconstructed to form a single smooth and seamless surface. We have intentionally presented our raw results after applying registration only so that the accuracy of our algorithm could be analyzed. In the future, we intend to use this algorithm for multi-view correspondence and registration. An extension of this work is to achieve multi-view correspondence without any prior knowledge of the ordering of the views.

6 Conclusion

We have presented a novel 3-D free-form object representation scheme based on tensors. We have also presented a fully automatic correspondence and registration algorithm, based on our novel representation. Our algorithm makes no assumption about the underlying surfaces and does not require initial estimates of registration or the viewing angles of the object. The strength of our algorithm lies in our robust representation scheme based on fourth order tensors. We have presented an effective and efficient procedure for matching these tensors to establish correct correspondence between a model and a scene surface. The algorithm has been tested on different data sets of varying mesh resolution and our results show the effectiveness of the algorithm.

Acknowledgments. We are grateful to the Robotics Institute, Carnegie Mellon University, USA for providing us with the data used in our experiments. This research is supported by ARC grant number DP0344338.

References

1. Mian, A. S., Bennamoun, M., Owens, R.: Automatic Correspondence for 3D Modeling: An Extensive Review. Submitted to a journal, (2004)
2. Besl, P.J., McKay, N.D.: Reconstruction of Real-world Objects via Simultaneous Registration and Robust Combination of Multiple Range Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2 (1992) 239–256
3. Chen, Y., Medioni, G.: Object Modeling by Registration of Multiple Range Images. IEEE International Conference on Robotics and Automation (1991) 2724–2729
4. Rangarajan, A., Chui, H., Duncan, J.S.: Rigid point feature registration using mutual information. Medical Image Analysis, Vol. 3, No. 4, (1999) 425–440
5. Chen, C., Hung, Y., Cheng, J. RANSAC-Based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 11 (1991) 1229–1234
6. Wyngaerd, J., Gool, L., Koth, R., Proesmans, M.: Invariant-based Registration of Surface Patches. IEEE International Conference on Computer Vision, Vol. 1 (1999) 301–306
7. Chua, C.S., Jarvis R.: 3D Free-Form Surface Registration and Object Recognition. International Journal of Computer Vision, Vol. 17, (1996) 77–99
8. Higuchi K., Hebert M., Ikeuchi K.: Building 3-D Models from Unregistered Range Images. IEEE International Conference on Robotics and Automation, Vol. 3, (1994) 2248–2253
9. Ashbrook, A.P., Fisher, R.B., Robertson, C., Werghi, N.: Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 2 (1998) 674–686
10. Stephens, R.S.: A probabilistic approach to the Hough transform. British Machine Vision Conference (1990) 55–59
11. Roth, G.: Registering Two Overlapping Range Images. IEEE International Conference on 3-D Digital Imaging and Modeling (1999) 191–200
12. Johnson, A.E., Hebert, M.: Surface Registration by Matching Oriented Points. International Conference on Recent Advances in 3-D Imaging and Modeling (1997) 121–128
13. Johnson, A.E.: Spin Images: A Representation for 3-D Surface Matching. PhD. Thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 (1997)
14. Foley, J., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics-Principles and Practice. Addison-Wesley, Second Edition (1990)
15. Zhang, Z.: Iterative Point Matching for Registration of Free-form Curves and Surfaces. International Journal of Computer Vision, Vol. 13, No. 2, (1994) 119–152