

From Unordered Range Images to 3D Models: A Fully Automatic Multiview Correspondence Algorithm

A. S. Mian, M. Bennamoun and R. A. Owens
School of Computer Science and Software Engineering
The University of Western Australia
35 Stirling Highway, Crawley, WA 6009, Australia
email: {ajmal, bennamou, robyn}@csse.uwa.edu.au

Abstract

In this paper, we present a fully automatic multiview correspondence algorithm based on our tensor representation. Our algorithm can automatically establish correspondences between a set of unordered range images of an object and register them in a common coordinate basis. Existing techniques of correspondence are either manual or inefficient and require the ordering of range images according to their overlap. Our algorithm does not suffer from any such limitation. We also present a complete and fully automatic 3D modeling framework by integrating our multiview correspondence algorithm with a global registration, an integration and a surface reconstruction algorithm. Our results show that our algorithm is fully automatic, applicable to free-form objects, efficient and independent of the resolution and sampling of the range images.

1. Introduction

A range image of a free-form object acquired from a single viewpoint contains occluded parts. In order to make a complete 3D model of the object, it must be acquired from multiple viewpoints. The number of range images required for a complete 3D model depends upon the complexity of the object and the level of detail required. These range images must be registered in a common coordinate basis. This is achieved by establishing correspondence between overlapping range images. Once the range images are registered in a single coordinate basis, they are integrated and reconstructed to form a complete 3D model which can then be viewed from all sides.

All the modules in the 3D modeling process have been automated except for correspondence. Correspondence is a difficult task because of the following reasons. First, there is no knowledge about the relative viewing angles of the different range images. Second, there is no knowledge about the order of overlap of the range images (which range im-

ages overlap and which ones do not). Third, given two overlapping range images, there is no knowledge about their region of overlap or the amount of overlap.

Existing techniques of correspondence like the ICP [3] algorithm, Chen and Medioni's algorithm [5] and registration by maximizing mutual information [17] are not fully automatic because they require initial estimates of registration. Correspondence techniques which claim to be automatic are inefficient and based on various assumptions. Techniques like the RANSAC-based DARCES [4] and graph matching [6] are inefficient because they are based on exhaustive search. The SAI matching [11] assumes the underlying surface to be free of topological holes. Spin image matching [13] requires a uniform sampling of the underlying surface, and its computational complexity increases with the increase in resolution of the range images. Roth's technique [18] requires a significant amount of texture to be present on the surface of an object. Geometric histogram matching [2] uses the Hough transform which is computationally expensive. Bitangent curve matching [22] extracts invariants from bitangent curves which are global features and may not be fully contained inside the overlapping region of the range images. It also requires the calculation of derivatives which are sensitive to noise. Three tuple matching [7] requires the calculation of first and second order derivatives which are sensitive to noise. This technique is also sensitive to the surface sampling and its computational complexity increases linearly with the increase in the resolution of the range images.

All the above algorithms are pairwise correspondence algorithms. None of these algorithms can solve the multiview correspondence problem, where the range images are unordered. To the best of our knowledge there is no automatic multiview correspondence technique available with the exception of Huber's 3D modeling framework [12]. However, this framework relies on an external algorithm for pairwise correspondence. For this purpose, it uses a modified version of the spin images which must find correspondences

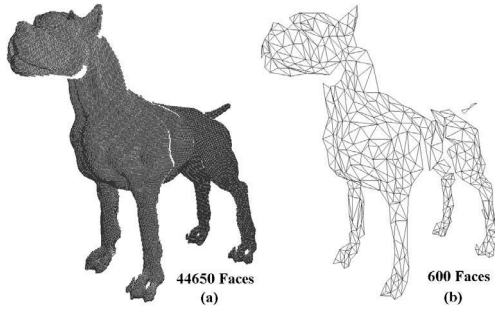


Figure 1. (a) A high resolution mesh of the dog. (b) After mesh reduction the number of facets is considerably reduced.

between $N(N-1)/2$ pairs of range images in order to generate a graph of relative pose estimates (N is the total number of range images). This has a computational complexity of $O(N^2)$ and is an additional overhead to the computational complexity of the rest of the framework.

In this paper we propose a fully automatic and efficient multiview correspondence algorithm based on our previous work [15] [16]. Our algorithm efficiently builds tensors for all range images and indexes them through a table for fast reference (Section 2). During the multiview correspondence, tensors of different range images at the same index position in the table are matched to derive a spanning tree graph containing range images as nodes and rigid transformations as arcs. Range images are removed from the search space as they are added to the graph in order to gain further efficiency. We also combine our automatic correspondence algorithm with other modular automatic components to make a complete automatic 3D modeling framework.

2. Tensor and Index Table Computation

During an offline representation phase, tensors are built for all the range images [15]. First all the range images, in the form of point clouds, are converted into triangular meshes M_i (where $i = 1, \dots, N$). For reasons of efficiency, a mesh reduction algorithm, that preserves maximum features on the surfaces, is applied to each mesh (see Fig. 1) resulting in reduced meshes M'_i . For this purpose, we used Garland's surface simplification algorithm [10]. However, any other efficient algorithm can be used. Normals are then calculated for each vertex of an M'_i . Next, sets of two points along with their normals are selected to define local 3D coordinate bases. To avoid the C_2^n (where n is the number of points in M'_i) combinatorial explosion of the points, a distance constraint is imposed on the pairing of the points. This distance constraint allows the pairing between only those points which are within a prespecified distance.

The allowable distance between point pairs is selected

as a fraction of the bounding dimensions of the object. All range images are transformed to their principal axes and the approximate x, y, z bounding dimensions ($\mathbf{D} = [D_x \ D_y \ D_z]$) of the object are calculated using Eqn. 1.

$$\mathbf{D} = \max_{xyz} (\max_{xyz} (\mathbf{V}_i \mathbf{P}_i) - \min_{xyz} (\mathbf{V}_i \mathbf{P}_i)) \quad (1)$$

In Eqn. 1, \mathbf{V}_i is an $n \times 3$ matrix of the x, y and z coordinates of the data points of i th range image ($i = 1, \dots, N$). \mathbf{P}_i is the rotation matrix which aligns \mathbf{V}_i along its principal axis. The operator " $\max_{xyz} (\mathbf{V}_i)$ " takes the maximum values of x, y , and z in \mathbf{V}_i .

The limits of the distance constraint (d_{min} and d_{max}) are calculated from \mathbf{D} using Eqn. 2 and Eqn. 3. The distance constraint ensures that the points that are selected to define the 3D basis are far enough so that the calculation of the coordinate basis is not sensitive to noise and close enough so that both points can lie inside the overlapping region. Each point is paired with only the closest three points which satisfy the distance constraint, limiting the maximum number of possible pairs to $3n$.

$$d_{min} = \frac{\text{mean}(D_x, D_y, D_z)}{6} \quad (2)$$

$$d_{max} = \frac{\text{mean}(D_x, D_y, D_z)}{4} \quad (3)$$

For each valid pair of points, a local 3D basis is defined as follows. The center of the line joining the two points defines the origin. The average of the two normals defines the z -axis. The cross product of the two normals defines the x -axis and finally the cross product of the z -axis with the x -axis gives the y -axis. This 3D basis (Fig. 2(a)) is used to define a 3D grid centered at the origin (Fig. 2(b)). Two parameters need to be selected, namely, the number of bins in the 3D grid and the size of each bin b_s . Varying the number of bins from less to more varies the representation from being local to being global. In our experiments we found that defining a $10 \times 10 \times 10$ grid gave good results. The bin size defines the level of granularity at which the object's surface is represented and is also calculated from \mathbf{D} using Eqn. 4.

$$b_s = \frac{\text{mean}(D_x, D_y, D_z)}{30} \quad (4)$$

Once the 3D grid is defined over the mesh, the surface area intersecting each bin of the grid is recorded in a third order tensor. Each element of the tensor is the area of the mesh that intersects the bin corresponding to that element inside the 3D grid (Fig. 2(d)). This tensor corresponds to a local representation of the surface. To find the area of intersection of the mesh with each bin of the 3D grid, we start from a point that is closest to the origin of the 3D basis and visit each triangular facet in its immediate neighbourhood. The area of intersection of a triangle and a bin is calculated using Hodgman's polygon clipping algorithm [9].

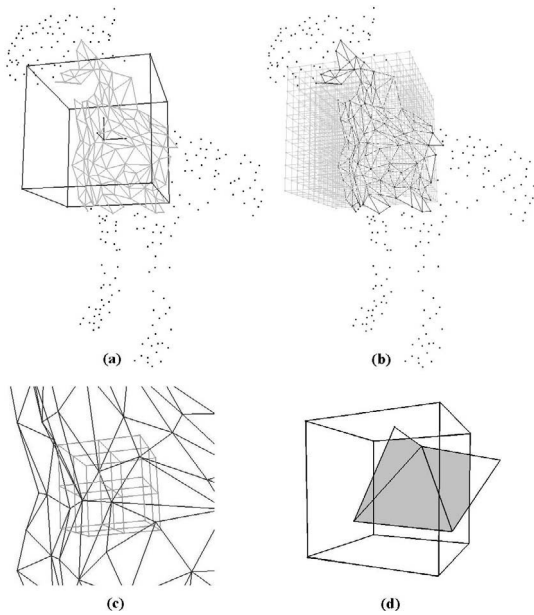


Figure 2. (a) 3D basis defined over the mesh of the dog. (b) A $10 \times 10 \times 10$ grid centered at the origin the 3D basis. (c) A zoomed in view of 8 bins around the origin. (d) A bin intersecting 3 facets. The area of intersection (shaded) is the value of the tensor element corresponding to this bin.

Once all the triangular facets in the immediate neighbourhood of the point have been visited and their intersection with the grid bins calculated, their next outer neighbouring triangular facets are considered. This process continues until a stage is reached when all the triangular facets are completely outside the 3D grid, at which point the computation is stopped. The angle between the normals of the two points defining a tensor, which we call the definition angle θ_{def} , is used in a 1D table to index the tensor for quick reference. Each bin of the index table serves as a quick reference to a group of tensors that are within a certain range ($\Delta\theta_{def}$) of θ_{def} . Choosing a small $\Delta\theta_{def}$ will reduce the number of possible matches for a tensor hence reducing matching time. However, it will also increase the risk of missing out a correct matching tensor. We found from our experiments that $\Delta\theta_{def} = 5^\circ$ gave good results.

Most of the bins of the 3D grid are likely to be empty (Fig. 2(b)) resulting in tensors with many zero entries. The tensors are converted into sparse arrays which reduces the memory utilization by approximately 85%. Moreover, tensors that have more than 95% elements as zero are discarded. Such tensors generally result from 3D grids defined over the boundaries of the views and are not likely to give correct matches. Discarding such tensors makes the match-

ing process more efficient in addition to the saving in memory.

3. Automatic Multiview Correspondence

The three tuple (M_i , M'_i and tensors) are fed to our automatic multiview correspondence algorithm. Our algorithm searches for a spanning tree graph containing range images as nodes and arcs representing transformation between its end nodes. Using this graph and by concatenating transformations, all the range images can be registered in the coordinates of a reference view. The algorithm proceeds as follows. The mesh M_R which has the maximum number of data points is selected as the root node to initialize the graph. Next, tensors of M_R are matched with the tensors from the remaining $N - 1$ meshes that are at the same index position in the index table. Details of the tensor matching are given in this section below. When a mesh is found with a matching tensor, it is added to the graph and joined with node M_R with an arc. The arc represents the transformation that aligns the two meshes. Once a mesh has been added to the graph, it is removed from the search list (see Fig. 3). Therefore, the remaining tensors of M_R would be matched with the tensors of $N - 2$ meshes only. This process continues until all those meshes whose tensor matches with a tensor of M_R are added to the graph. When all the tensors of M_R have been considered for matching, another mesh which has the highest number of data points is picked up from the graph and its tensors are matched with the remaining meshes in the search space. This process continues increasing the graph each time a match is found and decreasing the search space until all the meshes have been added to the spanning tree. Fig. 3 shows the trace of our algorithm for the 19 range images of the dog.

To match a tensor T_m of model mesh M'_m and tensor T_s of scene mesh M'_s , the following procedure is adopted. First, the overlap ratio R_O of the two tensors is calculated according to Eqn. 5 (where I_{ms} and U_{ms} are the intersection and union of the occupied bins of the two tensors). If R_O is greater than a threshold t_r (set to 0.5) the correlation coefficient C_c of the two tensors is calculated in their region of overlap (Eqn. 6). If C_c is greater than a threshold t_c (also set to 0.5), the algorithm proceeds to local-verification. During local-verification M'_s is transformed to the coordinates of M'_m to form M'_{ms} using the rotation matrix \mathbf{R} and the translation vector \mathbf{t} calculated with Eqn. 7 and Eqn. 8 respectively.

$$R_O = \frac{\sum I_{ms}}{\sum U_{ms}} \quad (5)$$

$$C_c = \text{correl coeff}(\mathbf{T}_m(I_{ms}), \mathbf{T}_s(I_{ms})) \quad (6)$$

$$\mathbf{R} = \mathbf{B}_s^\top \mathbf{B}_m \quad (7)$$

$$\mathbf{t} = \mathbf{O}_m - \mathbf{O}_s \mathbf{R} \quad (8)$$

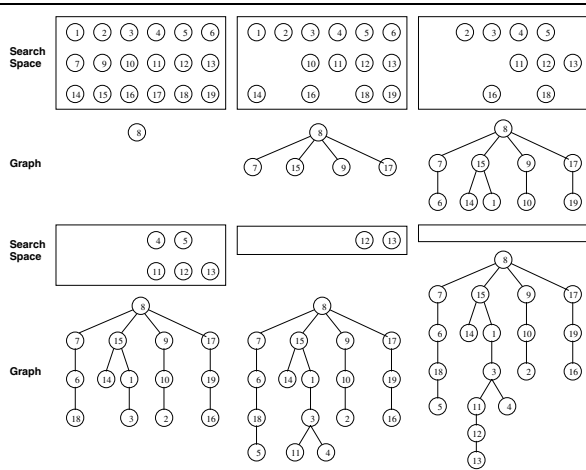


Figure 3. Trace of the multiview correspondence algorithm for the dog data set. The search space reduces as the graph is built.

In Eqn. 6, $\mathbf{T}_m(I_{ms})$ and $\mathbf{T}_s(I_{ms})$ are the tensor values in the region of overlap of \mathbf{T}_m and \mathbf{T}_s respectively. In Eqn. 7, \mathbf{B}_m and \mathbf{B}_s are the matrices of coordinate basis used to define the model and scene tensors respectively. \mathbf{B}_s^\top is the transpose of matrix \mathbf{B}_s . In Eqn. 8 \mathbf{O}_m and \mathbf{O}_s are the coordinates of the origins of \mathbf{T}_m and \mathbf{T}_s respectively (see Fig. 2(a)). \mathbf{M}'_{ms} is then aligned along its principal axis and its x, y, z bounding dimensions \mathbf{D}'_{ms} are calculated using Eqn. 9.

$$\mathbf{D}'_{ms} = \max_{xyz}(\mathbf{V}'_{ms}\mathbf{P}_{ms}) - \min_{xyz}(\mathbf{V}'_{ms}\mathbf{P}_{ms}) \quad (9)$$

In Eqn. 9, \mathbf{V}'_{ms} is the matrix of x, y and z coordinates of the data points of \mathbf{M}'_{ms} and \mathbf{P}_{ms} is the rotation matrix that aligns \mathbf{V}'_{ms} along its principal axis. In the next step, \mathbf{D} is subtracted from \mathbf{D}'_{ms} . If the maximum difference between the two is less than a prespecified tolerance t_D , \mathbf{M}_m and \mathbf{M}_s are also registered (with \mathbf{R} and \mathbf{t}) and pairs of points on \mathbf{M}_m and \mathbf{M}_s that are within a distance equal to $2d_{res}$ (d_{res} is the resolution of \mathbf{M}_i) are turned into correspondences. We selected a conservative value of $t_D = \text{mean}(D_x, D_y, D_z)/10$ since the registration at this stage is likely to be inaccurate. If the number of correspondences found is more than n_c ($n_c = \min(\text{number of points in } \mathbf{M}_m \& \mathbf{M}_s)/4$), the registration is refined with a variant of the ICP algorithm [19]. Correspondences are established once again between points on the two views that are within a distance of d_{res} . If the number of correspondences found is more than $2n_c$, the combined bounding dimensions \mathbf{D}_{ms} of the registered fine meshes are calculated in a similar fashion using Eqn. 9. Once again \mathbf{D} is subtracted from \mathbf{D}_{ms} . If the maximum difference between the two is less than a tolerance $2d_{res}$, the algorithm proceeds to the

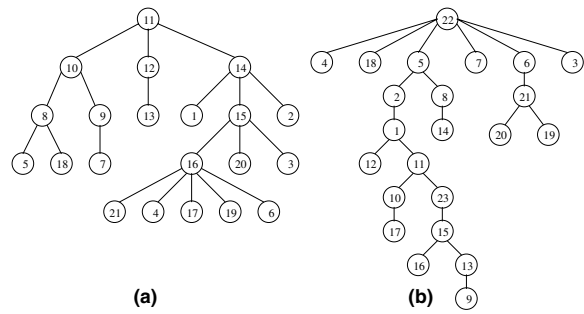


Figure 4. Correspondence graph of the range images of the bone (a) and the dinosaur (b).

global-verification step. If any one of the above local-verification steps fails, the next pair of tensors is selected for matching and the process is repeated.

Global-verification is performed before a new mesh can be added to the graph as follows. The combined bounding dimensions \mathbf{D}_L of the existing meshes in the spanning tree and the new mesh are calculated after they have been registered in a similar way using Eqn. 9. Next, \mathbf{D} is subtracted from \mathbf{D}_L . If the maximum difference between the two is less than $4d_{res}$, the newly added mesh is accepted in the graph. If global-verification fails, the whole process is repeated.

4. Completing the 3D Model

Once the correspondence graph is complete, all the meshes are registered in the coordinate basis of a reference mesh and correspondences are established between the close points of all overlapping meshes (not just the ones connected in the graph). This exhaustive list of correspondences is then fed to a global registration algorithm [21] which registers the views globally, distributing the registration errors evenly over the 3D model. Finally, the meshes are automatically integrated and reconstructed using Vrip-Pack [1] which uses the volumetric integration algorithm by Curless and Levoy [8] for integration and the marching cubes algorithm [14] for reconstruction.

5. Results

We performed our experiments on a large data set of high resolution range images (approx. 40,000 points per view). In this paper we present the results of our experiments performed on three objects namely, the dog, the dinosaur and the bone (range data taken from [20]). In each case the unordered range images of the object were fed to our algorithm and their respective 3D models were generated. Fig. 3 shows the spanning tree built from the range images of the dog whereas Fig. 4(a) and Fig. 4(b) show the spanning tree built from the range images of the bone and the dinosaur

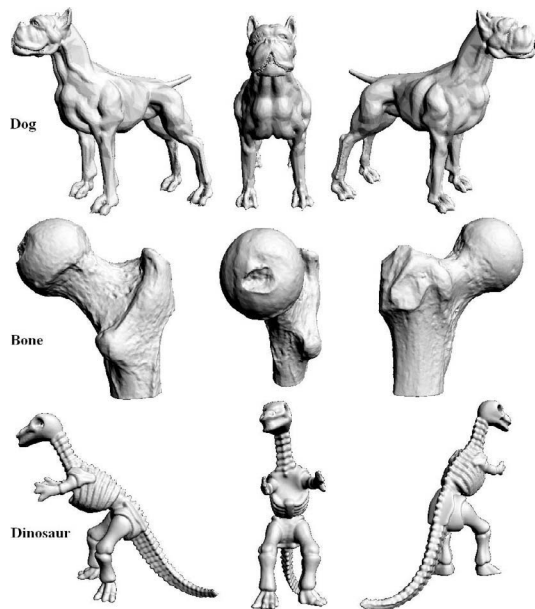


Figure 5. 3D modeling results.

respectively. Fig. 5 shows the 3D models of the objects viewed from three different angles.

6. Conclusion

We presented a fully automatic multiview correspondence algorithm based on our previously published tensor representation for registering unordered range images. Our algorithm uses an index table to efficiently match tensors of different views for establishing correspondences. We also integrated our algorithm with other modular automatic components to make a complete automatic 3D modeling framework. Our results show that our algorithm is fully automatic, efficient and applicable to free-form objects.

7. Acknowledgment

We are grateful to Carnegie Mellon University, USA for providing the mesh reduction software, to Stanford University, USA for providing VripPack software and to The University of Stuttgart, Germany for providing range data. This research is sponsored by ARC grant number DP0344338.

References

- [1] A Volumetric Range Image Processing Package. Stanford Computer Graphics Laboratory. <http://graphics.stanford.edu/software/vrip/>, Aug 2001.
- [2] A. Ashbrook, R. Fisher, C. Robertson, and N. Werghi. Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms. *IJPRAI*, 2:674–686, 1998.
- [3] P. Besl and N. McKay. Reconstruction of Real-world Objects via Simultaneous Registration and Robust Combination of Multiple Range Images. *IEEE TPAMI*, 14(2):239–256, February 1992.
- [4] C. Chen, Y. Hung, and J. Cheng. RANSAC-Based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images. *IEEE TPAMI*, 21(11):1229–1234, November 1991.
- [5] Y. Chen and G. Medioni. Object Modeling by Registration of Multiple Range Images. In *IEEE ICRA*, pages 2724–2729, April 1991.
- [6] J. Cheng and H. Don. A Graph Matching Approach to 3-D Point Correspondences. *IJPRAI*, 5(3):399–412, 1991.
- [7] C. Chua and R. Jarvis. 3D Free-Form Surface Registration and Object Recognition. *IJCV*, 17:77–99, 1996.
- [8] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In *SIGGRAPH*, volume 30, pages 303–312, 1996.
- [9] J. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics-Principles and Practice*. Addison-Wesley, Second Edition, 1990.
- [10] M. Garland and C. Heckbert. Surface Simplification Using Quadric Error Metrics. In *SIGGRAPH*, 1997.
- [11] K. Higuchi, M. Hebert, and K. Ikeuchi. Building 3-D Models from Unregistered Range Images. In *IEEE ICRA*, volume 3, pages 2248–2253, May 1994.
- [12] D. Huber and M. Hebert. 3D Modeling Using a Statistical Sensor Model and Stochastic Search. In *IEEE CVPR*, pages 858–865, 2003.
- [13] A. E. Johnson and M. Hebert. Surface Registration by Matching Oriented Points. In *IEEE Int. Conf. on Recent Advances in 3-D Imaging and Modeling*, pages 121–128, 1997.
- [14] W. Lorenzen and H. Cline. A High Resolution 3D Surface Construction Algorithm. In *ACM SIGGRAPH*, pages 163–169, 1987.
- [15] A. S. Mian, M. Bennamoun, and R. A. Owens. Matching Tensors for Automatic Correspondence and Registration. In *ECCV*, May 2004. To appear.
- [16] A. S. Mian, M. Bennamoun, and R. A. Owens. Performance Analysis of an Improved Tensor Based Correspondence Algorithm for Automatic 3D Modeling. In *IEEE ICIP*, October 2004. Submitted.
- [17] A. Rangarajan, H. Chui, and J. Duncan. Rigid Point Feature Registration Using Mutual Information. *Medical Image Analysis*, 3(4):425–440, 1999.
- [18] G. Roth. Registering Two Overlapping Range Images. In *IEEE Int. Conf. on 3-D Digital Imaging and Modeling*, pages 191–200, 1999.
- [19] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *3DIM*, pages 145–152, 2001.
- [20] Stuttgart Range Image Database. The University of Stuttgart. <http://range.informatik.uni-stuttgart.de/htdocs/html/>.
- [21] J. Williams and M. Bennamoun. Simultaneous Registration of Multiple Corresponding Point Sets. *CVIU*, 81(1):117–142, 2001.
- [22] J. Wyngaerd, L. Gool, R. Koth, and M. Proesmans. Invariant-based Registration of Surface Patches. In *IEEE ICCV*, volume 1, pages 301–306, 1999.