International Journal of Computer Vision 66(1), 19–40, 2006

DOI: 10.1007/s11263-005-3221-0

A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images*

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
ajmal@csse.uwa.edu.au
bennamou@csse.uwa.edu.au
robyn@csse.uwa.edu.au

Received September 7, 2004; Revised March 3, 2005; Accepted March 3, 2005

Abstract. Automatic registration of range images is a fundamental problem in 3D modeling of free-from objects. Various feature matching algorithms have been proposed for this purpose. However, these algorithms suffer from various limitations mainly related to their applicability, efficiency, robustness to resolution, and the discriminating capability of the used feature representation. We present a novel feature matching algorithm for automatic pairwise registration of range images which overcomes these limitations. Our algorithm uses a novel tensor representation which represents semi-local 3D surface patches of a range image by third order tensors. Multiple tensors are used to represent each range image. Tensors of two range images are matched to identify correspondences between them. Correspondences are verified and then used for pairwise registration of the range images. Experimental results show that our algorithm is accurate and efficient. Moreover, it is robust to the resolution of the range images, the number of tensors per view, the required amount of overlap, and noise. Comparisons with the spin image representation revealed that our representation has more discriminating capabilities and performs better at a low resolution of the range images.

Keywords: correspondence, automatic registration, feature matching, shape descriptor, 3D representation, 3D modeling

1. Introduction

Three dimensional modeling has many applications in computer graphics, virtual reality, medical science, reverse engineering and robotics. Various techniques including stereo, structured light and laser range finders are used for acquiring range images of an object. A range image (also known as a $2\frac{1}{2}$D image) is generally in the form of a point cloud (see Fig. 2(a)). A single range image (or a view) however is not sufficient to completely model a free-form object (Besl, 1990) due to self-occlusions. To complete the 3D model, multiple views of the object must be acquired in order to cover the entire surface of the object. These views must then be registered in a common coordinate basis. According to the survey of Campbell and Flynn (2001), registration is performed in two steps. First, the views are coarsely registered and second, the registration is refined with a fine registration algorithm like ICP (Besl and McKay, 1992) for example.

Coarse registration can be performed either manually or automatically. In the former approach, corresponding points are manually identified in the

*This work has been provisionally patented under Australian patent number 2004902436 and is sponsored by ARC grant number DP0344338.
†To whom correspondence should be addressed.

overlapping region of two different views. 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 a rigid transformation (rotation and translation) that aligns the views. Automatic coarse registration can be achieved in two different ways. One, by tracking the motion of the object (to be modeled) relative to the sensor and applying the reverse transformations to the views. This method either makes the sensing device expensive or limits its capability to only scanning small objects that can be placed on a turn table. The second way is through feature matching which is also known as automatic correspondence identification. Feature matching automatically identifies corresponding points in the two views and coarsely registers them by minimizing the distance between these points.

Coarse registration is followed by a fine registration algorithm which iteratively refines the initial coarse registration. The classic Iterated Closest Point (ICP) algorithm (Besl and McKay, 1992), the Chen and Medioni's algorithm (1991) and the registration approach based on maximizing mutual information (Rangarajan et al., 1999) are examples of fine registration algorithms. These algorithms can only work once the views have been coarsely registered. Moreover, in case the coarse registration is not accurate enough, these techniques may not converge to the correct solution. In addition to pairwise fine registration, more than two views can also be simultaneously registered with multiview global registration techniques (Williams and Bennamoun, 2001; Benjemma and Schmitt, 1997; Oishi et al., 2003; Nishino and Ikeuchi, 2002). Once all the views are registered in a common coordinate basis, they are integrated and reconstructed to make a complete 3D model. The block diagram of Fig. 1 illustrates the process of automatic 3D modeling.

The idea behind feature matching based automatic coarse registration techniques is to represent the features of each range image and match these representations in order to identify corresponding points between them. For accurate feature matching, ideally the representation used must be unique and invariant to rigid transformations. A unique representation should result in a similar representation only for exactly similar features. However, in practice features of an object acquired by different range images vary to some extent due to the variations caused by noise and surface sampling. Therefore, representations must be adapted to handle these variations. As a result, these representations no longer remain unique and loose some of their discriminating capability i.e. matching of these representations results in one-to-many matches including incorrect ones. The challenge here for a representation scheme is to be capable of handling small variations in features while still maintaining maximum discriminating capability between features. A representation with low discriminating capability will result in multiple ambiguous matches making it difficult to identify correct matches from incorrect ones. The end result is that the algorithm becomes computationally expensive and may even converge to an incorrect solution.

Various representation schemes have been used by feature matching algorithms for automatic coarse registration. However, these algorithms or the representations they use suffer from a number of limitations related to their applicability to free form objects, efficiency, robustness to resolution and low discriminating capability of the representation. The following is a survey of related work in the area of automatic coarse registration by feature matching. The RANSAC-based DARCES algorithm (Chen et al., 1991) is based upon an exhaustive search and is not practical if the data sets are large. Moreover, the DARCES algorithm makes some unrealistic assumptions about the overlapping regions of the views. Another example of an exhaustive search based algorithm is the graph matching algorithm (Cheng and Don, 1991). Bitangent curve matching (Wyngaerd et al., 1999) requires first order derivatives which are sensitive to noise. Another problem with bitangent curves is that they represent global features which may not be fully contained inside the

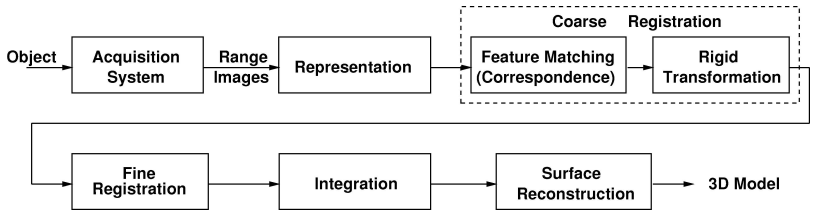

Figure 1. Block diagram showing the different components of a generic automatic 3D modeling system.

overlapping regions of the views. Three tuple matching (Chua and Jarvis, 1996) calculates the first and second order derivatives which are also sensitive to noise and require the underlying surfaces to be smooth. Spherical Attribute Image (SAI) matching (Higuchi et al., 1994) requires the underlying surfaces to be free of topological holes which limits its applicability. Geometric histogram matching (Ashbrook et al., 1998) makes use of a 3D Hough transform (Stephens, 1990) which is computationally expensive. Roth's technique (Roth, 1999) is limited by the fact that it 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 (Johnson and Hebert, 1997) uses the spin image representation which has a low discriminating capability (as illustrated in Section 8) because it maps the 3D range image into a 2D histogram. Spin image matching therefore results in many ambiguous correspondences which must be processed through a number of filtration stages to prune out incorrect ones making the technique computationally inefficient even for range images of a reasonable size.

In this paper, we present a novel feature matching algorithm for automatic coarse registration of range images. Our algorithm uses a novel tensor representation for representing semi-local surface patches of a range image of a free-form (sculpted) object (Besl, 1990). Each range image is represented with multiple third order tensors. Each tensor is derived by defining a local 3D grid over the range image and quantizing the surface area intersecting each bin of the grid in a third order tensor (see Section 2 for details). This results in a 3D representation of the surface patch with a high discriminating capability leading to correct matches.Tensors of two overlapping views are matched to establish pairwise correspondences between the views. Corresponding tensors are verified (`local-verification`) and then used to pairwise register the views. The registration is then refined with a fine registration algorithm as illustrated in Fig. 1.

We combined our automatic pairwise registration algorithm with other modular components to devise a complete framework for automatic 3D modeling from *ordered* range images i.e. when a priori knowledge of overlapping view pairs is available. However, no further information about the relative viewing angles or exact regions of overlap of the views is available. Our algorithm pairwise registers the views after `local-verification` (Sections 4.1). Next a `global-verification` (Section 4.2) of the registration is performed considering all the views that have already been pairwise registered. A pairwise registration is accepted only if it passes both verifications, otherwise it is rejected and another pair of matching tensors is sought. Once all the views are pairwise registered, the registration is refined with a global registration algorithm (Williams and Bennamoun, 2001) which registers the views globally, distributing the registration errors evenly over the entire 3D model. The views are finally integrated and reconstructed to form a smooth and seamless 3D model.

We performed the analysis of our automatic pairwise registration algorithm (Mian et al., 2004d) taking into consideration the following criteria: accuracy of registration, robustness to resolution and the number of tensors per view, efficiency with respect to memory and time, robustness to the required amount of overlap and finally robustness to noise. We also compared our algorithm to the spin image matching algorithm (Johnson and Hebert, 1997) by applying both algorithms to the same sets of range images. Our results show that our algorithm has more discriminating capability and performs better than the spin image algorithm at a low resolution of the range images.

The rest of this paper is organized as follows. In Section 2, we describe our novel tensor representation scheme. In Section 3, we analyze the stability of our tensor representation. In Section 4, we give details of our automatic pairwise registration algorithm. In Section 5, we briefly describe our framework of automatic 3D modeling. In Section 6, we present our 3D modeling results along with their qualitative analysis. In Section 7, we report on the quantitative results and analysis of our automatic pairwise registration algorithm according to our laid down criteria. In Section 8, we perform the comparative analysis of our algorithm with the spin image matching algorithm. In Section 9 we give our conclusions and directions for future work.

2. Tensor Representation

During the representation phase (see Fig. 1), the input views of the object are converted into their tensor representations. To compute these tensors, the range images, in the form of point clouds (Fig. 2(a)), are converted into triangular meshes $\mathbf{M}_i$ where $i = 1, \ldots, N$ (Fig. 2(b)). This is performed by mapping the 3D points onto the 2D retinal plane of the sensor and performing a 2D Delaunay triangulation over the mapped points.

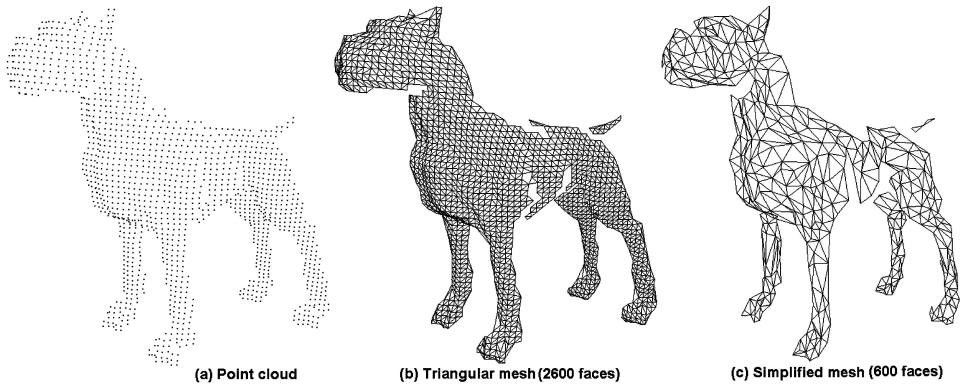

Figure 2. (a) A point cloud of a view of a dog (see Fig. 11 for a complete 3D model). (b) After converting into a triangular mesh. (c) After mesh reduction the number of faces are considerably reduced.

After triangulation, the points are mapped back to the 3D space and the triangles with edges longer than a prespecified threshold are removed. This separates surfaces which are falsely connected by the Delaunay triangulation. In our implementation we removed all triangles with an edge length 0.6 standard deviations longer than the mean.

For reasons of efficiency, a mesh reduction algorithm is applied to each mesh $\mathbf{M}_i$, resulting in a reduced mesh $\mathbf{M}'_i$ with approximately 400 vertices per mesh (see Fig. 2). For this purpose, we used Garland's mesh simplification algorithm (Garland and Heckbert, 1997). However, any other efficient algorithm can be used as long as it simplifies the mesh while preserving maximum amount of geometric variation on its surface. Normals are then calculated for each vertex of the reduced meshes using two different approaches. In the first approach, the normal of a vertex of $\mathbf{M}'_i$ is defined as the weighted average of the normals of its immediate neighbouring triangles. In the second approach, the normal of a vertex of $\mathbf{M}'_i$ is taken as the normal of its corresponding vertex in $\mathbf{M}_i$ which is defined as the weighted average of the normals of the triangles within a predefined neighbourhood of the vertex. The first approach is fast whereas the second approach gives more stable normals. However, our experiments show that both approaches give similar registration results given that the registration is refined with a fine registration algorithm.

Once the normals have been calculated, pairs of vertices along with their normals are selected to define local 3D coordinate bases. Note that a 3D coordinate basis can also be defined using three vertices without their normals. However, in this case the number of possible vertex pairs will be C_3^n (where n is the number of vertices in $\mathbf{M}'_i$) as opposed to the C_2^n possible pairs in the case of using two vertices and their normals. Moreover, choosing two vertices for defining a 3D coordinate basis also increases the chances that all vertices in a pair will belong to the region of overlap of the two meshes. Therefore, we define local coordinate basis on a mesh using two vertices and their normals.

To avoid the C_2^n combinatorial explosion of the vertex pairs, a distance constraint is imposed on their pairing. This distance constraint allows the pairing between only those vertices which are within a prespecified distance. The distance constraint also ensures that the vertices that are paired are far enough apart so that the calculation of the coordinate bases is not sensitive to noise but close enough to maximize their chances of being inside the overlapping region. The allowable distances between vertex pairs is selected as a fraction of the dimensions of the object. To calculate the dimensions of the object, all its range images are transformed to their principal axes in order to align their maximum surface area along the xy plane. The approximate bounding dimensions ($\mathbf{D} = [D_x\ D_y\ D_z]$) of the object along the x, y, z directions are then calculated using Eq. (1).

$$\mathbf{D} = \max_{xyz}\Big(\max_{xyz}(\mathbf{V}_i\mathbf{P}_i) - \min_{xyz}\mathbf{V}_i\ \mathbf{P}_i)\Big) \quad \forall\, i \in [1, \ldots, N] \tag{1}$$

In Eq. (1), $\mathbf{V}_i$ is an $n \times 3$ matrix of the x, y, z coordinates of the data points of the ith view. $\mathbf{P}_i$ is the rotation matrix which aligns $\mathbf{V}_i$ with its principal axis. The operator "$\max_{xyz}(\mathbf{V}_i)$" takes the maximum values of x, y, z in $\mathbf{V}_i$. If the views of an object completely cover its surface, then $\mathbf{D}$ is approximately equal to the

bounding box of the object when it is aligned with its principal axis. **D** serves as a low cost verification of the registration of two or more views. According to this verification step, the combined bounding dimensions of any number of registered views, when aligned with their principal axis, should not exceed **D**. Details of the verification are given in Sections 4.1 and 4.2.

The minimum and maximum limits ($d_{\min}$ and $d_{\max}$ respectively) of the distance constraint for pairing vertices are calculated from the bounding dimensions **D** of the object using Eq. (2).

$$\begin{aligned} d_{\min} &= \frac{\text{mean}(D_x, D_y, D_z)}{6} \\ d_{\max} &= \frac{\text{mean}(D_x, D_y, D_z)}{4} \end{aligned} \tag{2}$$

In addition to the distance constraint, an angle constraint is also imposed on the pairing of vertices so that vertices with approximatelyequal normals are not paired (since their cross product will result in a zero). According to the angle constraint, the normals of two vertices in any pair must have a mutual angle (θ_d) greater than 5°. Moreover, the average of the two normals must be defined i.e. when the two normals are added they must not result in a zero. Each vertex is paired only with its closest three vertices that satisfy the above constraints, limiting the maximum number of possible pairs to $3n$ per view (where n is the number of vertices per view). In practice the valid number of vertex pairs is much less than $3n$ due to the distance and angle constraints.

For each valid pair of vertices a local 3D basis is defined as follows. The center of the line joining the two vertices defines the origin of the new 3D basis. 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 defines the y-axis. This 3D basis is used to define a 3D grid centered at its origin (Fig. 3(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 gives good results (see Sections 6 and 7). The bin size defines the level of granularity at which the object's surface is represented. In our initial experiments (Mian et al., 2004), we defined the bin size in terms of the mesh resolution. However, with the introduction of an additional step of mesh reduction which results in a mesh with extremely non-uniform resolution, the bin size is now automatically calculated from the bounding dimensions of the object (Eq. (3)).

$$b_s = \frac{\text{mean}(D_x, D_y, D_z)}{30} \tag{3}$$

Once the 3D grid is defined, the surface area of the mesh intersecting each bin of the grid is recorded in a third order tensor. In simple terms, the tensor can be considered as a $10 \times 10 \times 10$ array of scalar elements where each element is the area of intersection between the mesh and the bin inside the 3D grid which corresponds to the same index location as the tensor element (Fig. 3(d)). This tensor is a local surface descriptor which corresponds to a local representation of the surface inside the 3D cubic grid. To find the area of intersection of the mesh with each bin of the 3D grid, we start from a vertex on the mesh that is closest to the origin of the 3D basis and visit each triangular facet in its immediate neighbourhood. We call this the `current-neighbourhood`. A single triangular facet may intersect more than one bin (Fig. 3(d)). The area of intersection of each triangular facet (in the `current-neighbourhood`) with its intersecting bins is calculated using Sutherland Hodgman's polygon clipping algorithm (Foley et al., 1990) and an entry is made at the corresponding element position in the tensor. Since more than one triangular facet can intersect a single bin, the calculated area of intersection is added to the area already present in that bin as a result of its intersection with another triangular facet. Once all the triangular facets have been visited in the `current-neighbourhood`, it becomes the `old-neighbourhood`. The "outer" (with respect to the origin) neighbouring triangular facets of the `old-neighbourhood` make up the new `current-neighbourhood`. The area of intersection of each triangular facet in the new `current-neighbourhood` with the grid bins is calculated as discussed above and entered into the tensor. This process continues until a stage is reached when all the triangular facets in the `current-neighbourhood` are completely outside the 3D grid at which point the computation is stopped. Note that there are chances that a surface may re-enter the grid after leaving it. Dealing with such situations may require looking for polygons in every bin of the grid which is computationally more expensive than the above region growing algorithm. Luckily, the chances

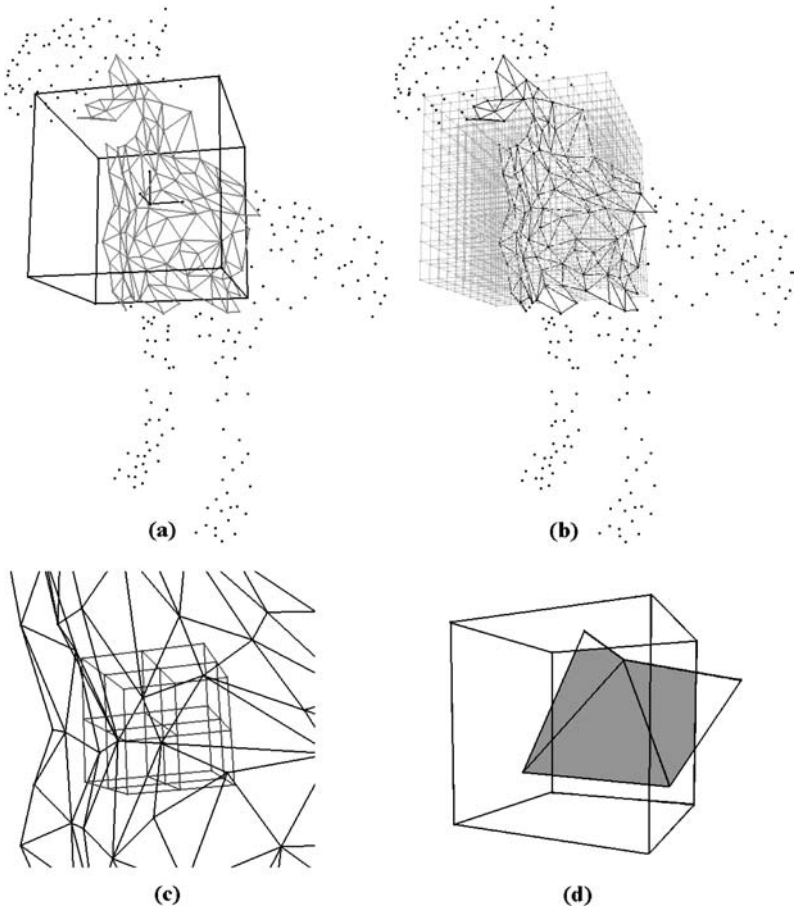

Figure 3. (a) 3D basis defined over the surface of the dog. (b) A $10 \times 10 \times 10$ grid defined over the surface of the dog centered at the origin of the 3D basis. The surface area of the mesh intersecting each bin ofthe grid is the value of the tensor element corresponding to the bin. (c) A zoomed in view of eight bins around the origin. (d) A single bin intersecting three triangular faces. The shaded area is the intersection of the faces with the bin. (Figure reproduced from Mian et al., 2004b.)

of such situations are very slim and therefore the region growing is a preferred approach.

Since most of the bins of the 3D grid are likely to be empty (see Fig. 3(b)), the resultant tensor will have many zero elements. In order to improve memory utilization, the tensor is compressed to a sparse form by squeezing out the zero elements and retaining the non-zero elements and their index positions in the third order tensor. This process reduces the memory utilization by approximately 85%. Compressed tensors are calculated for all the valid vertex pairs of each mesh in a similar way. These compressed tensors together with their respective coordinate basis and θ_d are called the tensor representation of the mesh or view. Tensors of each mesh are indexed by a 1D table with their θ_d for quick reference. Each bin of the index table serves as a quick reference to a group of tensors which have a θ_d within a certain range $\Delta\theta_d$. Choosing a small $\Delta\theta_d$ will reduce the number of possible matches for a tensor. However, it will also increase the risk of missing out a correct matching tensor due to noise in the vertex normals. We found from our experiments that a $\Delta\theta_d = 5^\circ$ gives good results (Mian et al., 2004b). Since the mutual relationship between the two vertices in a pair and their normals is invariant to rigid transformations, the 3D coordinate basis and the resulting tensors also have the same property.

3. Stability Analysis of the Coordinate Frames

Tensors derived from different views of the same surface will be similar or matching (see Section 4 for the similarity measure and matching of tensors) if they are

calculated with respect to similar coordinate frames. Therefore, it is importantto analyze the stability of the defined local 3D coordinate frames used to calculate our tensors. Stability here refers to the probability of getting similar coordinate frames in the overlapping regions of the meshes of two views of an object. The local coordinate basis $\mathbf{B}_1$ on view 1 and $\mathbf{B}_2$ on view 2 (each derived from a pair of vertices of the respective view) are considered similar if they satisfy the condition of Eq. (4).

$$\mathrm{B}_2^{\top}\mathrm{B}_1 \approx \mathrm{R}_{GT}^{-1} \tag{4}$$

$\mathbf{B}_i (i = 1, 2)$ is a 3×3 matrix of x, y, z coordinate vectors of the local coordinate basis of view i with respect to the view coordinate basis of view i and $\mathbf{R}_{GT}$ is the ground truth rotation matrix between view 2 and view 1. Two coordinate frames are likely to be similar if the vertex pairs used to derive them are in approximately the same location on the underlying object's surface. In other words, two coordinate frames will be similar if they are derived from corresponding pairs of vertices (see Section 1 for the definition of correspondence). Therefore, the stability of our coordinate frames is directly related to the probability of having corresponding pairs of vertices in overlapping meshes after mesh simplification (since the vertex pairs for defining a 3D basis are selected from the simplified meshes).

We performed a simple experiment to estimate the probability of corresponding vertices in two overlapping and simplified meshes. We took two meshes (at approximately 23000 vertices permesh) of two different views of the dog and calculated the percentage of vertices in mesh 2 which had a corresponding vertex in mesh 1 after fine registration. A pair of vertices (one from each mesh) is considered corresponding if the distance between them is less than 1.4 cm (namely twice the mesh resolution before simplification). This figure came out to be approximately 60%.[1] The two meshes were simplified (Garland and Heckbert, 1997) to approximately 270 vertices per mesh (or 400 faces per mesh) (see Fig. 4(a) and (b)) and the percentage of vertices of mesh 2 that were within a distance of 1.4 cm of view 1 was calculated again (see Fig. 4(c)). This time it came out to be 41.8% which is a reduction by only 18.2%. Assuming that the probability of finding a pair of corresponding vertices in the region of overlap of the two high resolution meshes is 1, after mesh simplification this probability will reduce to 0.7. This reduction is very reasonable when compared to a mesh simplification by a factor of $\frac{1}{85}$ or 98.8% reduction (i.e. 98.8% of the vertices are removed as a result of the mesh reduction). From Fig. 4(c) we can see that the vertices of the two meshes are very close in their region of overlap. Figure 4(d) shows a histogram of the distances between the vertices of mesh 2 and their nearest neighbour vertices in mesh 1. Figure 4(d) shows that a high percentage of the vertex pairs are closely located whereas some of the pairs have large distances between them mainly because at least one vertex of the pair does not belong to the region of overlap. We repeated the same experiment for three other objects namely the dinosaur, the bone and the dragon (their full models are shown in Fig. 11). Figure 5 shows our results calculated in exactly the same manner as in the case of the dog. From Fig. 5(a), (b) and (c), we can see that most of the vertices of mesh 2 have a closely located vertex in mesh 1 in each case. The statistics reported in Fig. 5 also reveal that only a small reduction in correspondences occurred in each case. Note that a threshold of 1.4 cm for corresponding vertices is extremely conservative compared to the dimensions of these objects. For example, by increasing this threshold to 2.8 cm there is almost no reduction in the percentage of corresponding vertices after mesh simplification (see Figs. 4(d) and 5). Our results show that the vertices of the simplified meshes of overlapping views of an object are closely located. Since the normal of a vertex is related to its position, the corollary is that these vertices are also likely to have similar normals especially when the normals are calculated using the high resolution mesh $\mathbf{M}_i$. On this basis, we conclude that the local coordinate frames derived from the corresponding pairs of two overlapping meshes will be approximately similar. Minor variations in the coordinate frames do not cause any problem because these coordinate frames are used only for automatic coarse registration. This coarse registration is then refined with a fine registration algorithm which is independent of these coordinate frames. Our results reported in Sections 6, 7 and 8 also support our argument.

The Garland and Heckbert's mesh simplification approach is viewpoint independent and produces similar sets of vertices for surfaces which are exactly similar. Rotating an acquired surface around the viewing direction (or any other axis) does not affect its simplification since the surface remains the same. However, when the viewing direction of a sensor is changed, it may acquire different parts (views) of a surface of an

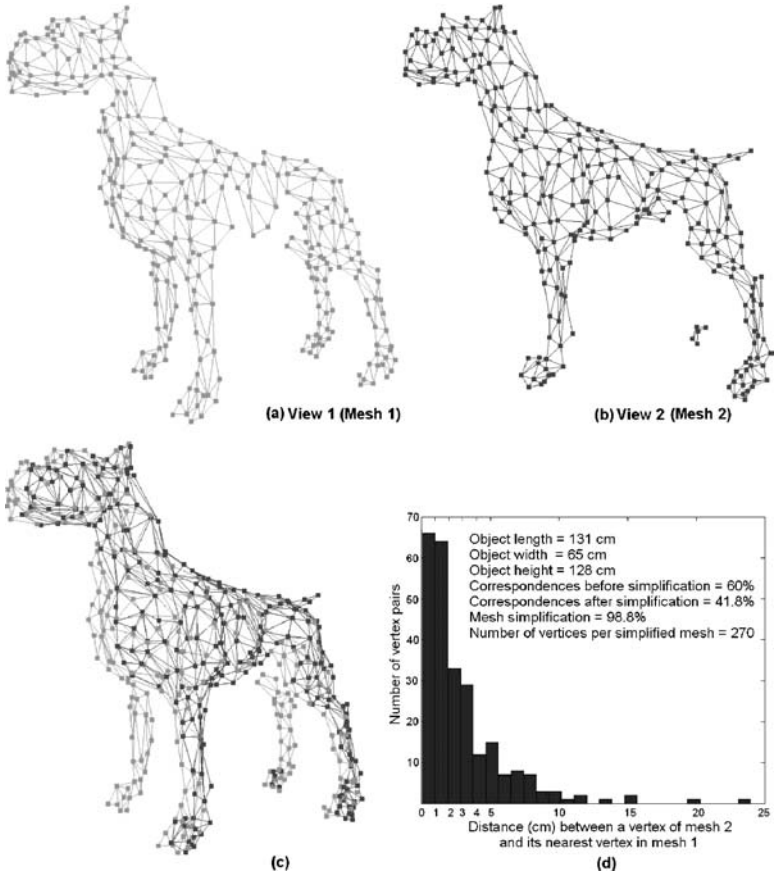

Figure 4. (a) and (b) Simplified meshes of two different views of the dog (see Fig. 11 for a complete 3D model) at approximately 270 vertices (or 400 faces) per mesh. (c) The two meshes are registered with a transformation calculated from the fine registration of the high resolution meshes. Note that the vertices of the two meshes are closely located. (d) A histogram of the distances between the vertices of mesh 2 and their nearest vertices in mesh 1. Most of the vertex pairs have a small distance between them whereas some of the vertex pairs have a large distance mainly because at least one vertex of the pair doesn't belong to the region of overlap of the two meshes. (*Note*: this figure is best viewed in colour.)

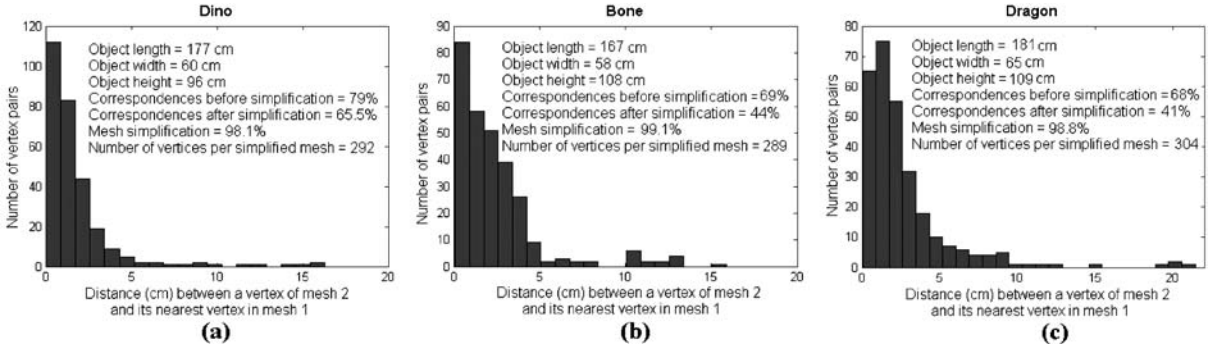

Figure 5. Histograms of the distances between the vertices of view 2 and their nearest vertices in view 1 of (a) the dinosaur, (b) the bone and (c) the dragon. (See Fig. 11 for the 3D models of these objects). The mesh resolution in each case was approximately 0.7 cm before simplification. The distance threshold for a corresponding pair of vertices was 1.4 cm in each case. Most of the vertex pairs have a small distance between them in each case compared to the dimensions of the object.

object. If these two views have sufficient overlap, their simplification will result in approximately similar sets of vertices in their region of overlap (see Fig. 5). This however cannot be guaranteed when the overlap between the two views is very small, as the tessellation of the overlapping region will be influenced by the rest of the meshes. Our results (Section 7.4) show that at least 50% overlap between the views is required by our algorithm for a successful registration. For further details about the Garland's mesh simplification algorithm and its error analysis, the reader is referred to Garland (1999).

4. Matching Tensors for Automatic Pairwise Correspondence and Registration

Tensors of a pair of views are matched to find correspondences between the views. Matching tensors are then used for the automatic coarse registration of the views by aligning the 3D coordinate basis used to derive these tensors. Since a tensor is a 3D descriptor of a local surface patch of anobject, a pair of matching tensors reveals that the surface patches represented by these tensors are also similar and should correspond to the same surface patch of the object. We use a linear correlation coefficient for matching a pair of tensors. Figure 6 shows the histogram of matches between a tensor of one view with 700 tensors of another view. It is clear from Fig. 6 that the matching pairs of tensors have a very high correlation value (as defined below) compared to the remaining tensors.

We will use hereafter the terminology of model view (reference view) and scene view instead of view 1 and view 2 for explaining our algorithm. To establish correspondence between a model mesh $\mathbf{M}_m$ and scene mesh $\mathbf{M}_s$ (where $m, s = 1, 2, \ldots, N$), a tensor is selected from $\mathbf{M}_s$ and matched with only those tensors of $\mathbf{M}_m{}'$ which are at $\theta_d \pm \Delta\theta_d$ positions in the index table. The matching of a scene tensor $\mathbf{T}_s$ and a model tensor $\mathbf{T}_m$ proceeds as follows. First, the overlap ratio R_O of the two tensors is calculated using Eq. (5).

$$R_O = \frac{n_q}{n_m + n_s - n_q} \tag{5}$$

In Eq. (5), n_q is the number of non-zero elements of $\mathbf{T}_m$ which have a corresponding non-zero element at the same index position in $\mathbf{T}_s$. In other words, n_q is the number of intersecting bins between $\mathbf{T}_m$ and $\mathbf{T}_s$. n_m and n_s are the total number of non-zero elements of $\mathbf{T}_m$ and $\mathbf{T}_s$ respectively. If R_O is greater than a prespecified threshold t_r, the algorithm proceeds to calculate the correlation coefficient C_c of the two tensors in their region of overlap Eq. (6), otherwise the next tensor from $\mathbf{M}_m{}'$ is considered for matching. In our experiments we found that $t_r = 0.5$ gives good results (see Sections 6 and 7). The tensors are matched only in those bins where both tensors have surface data to cater for situations where some part of the object may be occluded in one view.

$$C_c = \frac{n_q \sum_{i=1}^{n_q} p_i q_i - \sum_{i=1}^{n_q} p_i \sum_{i=1}^{n_q} q_i}{\sqrt{n_q \sum_{i=1}^{n_q} p_i^2 - \left(\sum_{i=1}^{n_q} p_i\right)^2}\sqrt{n_q \sum_{i=1}^{n_q} q_i^2 - \left(\sum_{i=1}^{n_q} q_i\right)^2}} \tag{6}$$

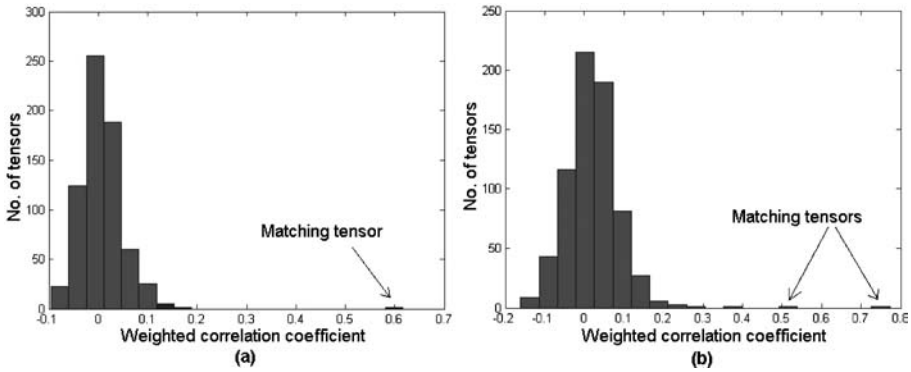

Figure 6. Histograms of matches (correlation coefficient weighted by the number of tensor elements n_q used to calculate the correlation coefficient) between a tensor of view 1 with 700 tensors of view 2. The matching tensors in each case have a very high correlation value compared to the remaining tensors. Note however that in both cases, these matching tensors will be verified as per Sections 4.1 and 4.2.

The Automatic Correspondence and Registration Algorithm

Input : $\mathbf{M}_i, \mathbf{M}'_i$ *(where* $i = 1, 2, \ldots, N$*), Tensors*
Output : *Transformations, Correspondences*

1. **for** each pair of overlapping views $\mathbf{M}_m$ and $\mathbf{M}_s$
2. **for** each tensor $\mathbf{T}_m$ of $\mathbf{M}_m$ and $\mathbf{T}_s$ of $\mathbf{M}_s$
3. **if** $R_O \geq t_r$ **AND** $C_c \geq t_c$
4. Transform $\mathbf{M}'_s$ to $\mathbf{M}'_m$
5. Calculate bounding dimensions $\mathbf{D}'_{ms}$ of $\mathbf{M}'_{ms}$
6. **if** max $(\mathbf{D}'_{ms} - \mathbf{D}) \leq t_D$
7. Transform $\mathbf{M}_s$ to $\mathbf{M}_m$
8. Establish correspondences between $\mathbf{M}_m$ and $\mathbf{M}_s$
9. **if** the number of correspondences $\geq n_c$
10. Refine registration with ICP
11. Establish correspondences between $\mathbf{M}_m$ and $\mathbf{M}_s$
12. **if** the number of correspondences $\geq 2n_c$
13. Calculate bounding dimensions $\mathbf{D}_{ms}$ of $\mathbf{M}_{ms}$
14. **if** max $(\mathbf{D}_{ms} - \mathbf{D}) \leq 2d_{res}$
15. Find dimensions $\mathbf{D}_L$ of the L views registered so far
16. **if** max $(\mathbf{D}_L - \mathbf{D}) \leq 4d_{res}$
17. **RETURN** Transformation and Correspondences

Figure 7. Pseudo-code of the automatic correspondence and registration algorithm. This corresponds to module 4 of the block diagram in Fig. 9. (Figure reproduced from Mian et al., 2004d).

In Eq. (6), p_i (i =1 ... n_q) are the elements of $\mathbf{T}_m$ in the region of overlap (intersection) of $\mathbf{T}_m$ and $\mathbf{T}_s$ i.e. the elements of $\mathbf{T}_m$ which have a corresponding element in $\mathbf{T}_s$ at the same index position. Similarly, q_i are the elements of $\mathbf{T}_s$ in the overlapping region of the two tensors. Note that $i = 1 \ldots n_q$ in both cases. If C_c is greater than a prespecified threshold t_c (which is also set to 0.5), the algorithm proceeds to the next step of `local-verification`.

4.1. *Local Verification (Steps 4 to 14 of the Pseudo-Code of Fig. 7)*

During `local-verifications`, all the points of $\mathbf{M}_s{}'$ are transformed to the coordinates of $\mathbf{M}_m{}'$. This transformation is calculated by transforming the corresponding 3D basis of $\mathbf{T}_s$ to the 3D basis of $\mathbf{T}_m$ using Eqs. (7) and (8).

$$\mathbf{R} = \mathbf{B}_\mathbf{s}^\top \mathbf{B}_\mathbf{m} \tag{7}$$

$$\mathbf{t} = \mathbf{O}_s - \mathbf{O}_m \mathbf{R} \tag{8}$$

$\mathbf{B}_i$ ($i = m, s$) is a 3×3 matrix of the x, y, z coordinate vectors (with respect to the view coordinate basis of view i) of the local coordinate basis used to derive $\mathbf{T}_i$. $\mathbf{O}_i$ is a 1×3 vector of the coordinates of the origin of $\mathbf{B}_i$ in the view coordinate basis of view i. $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector respectively which aligns $\mathbf{M}'_s$ with $\mathbf{M}'_m$ (the scene view with the model view). Figure 8 shows automatic pairwise coarse registration of meshes on the basis of a single pair of matching tensors. Note that our coarse registration results in Fig. 8 are quite accurate even though no registration refinement has been performed at this stage.

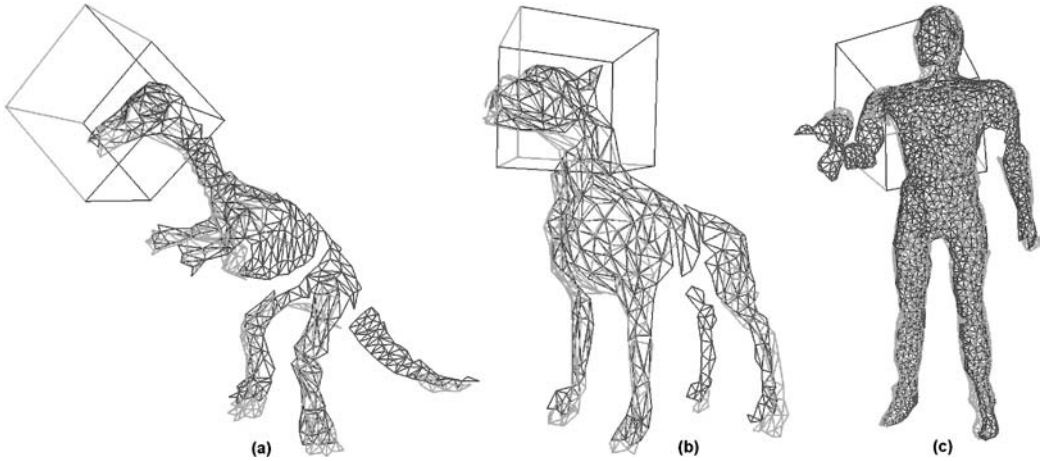

Figure 8. Automatic coarse registration of a pair of views of (a) the dinosaur, (b) the dog and (c) the robot. The cube in each case shows the bounding box of the 3D grid inside which the tensors were computed. Note that our coarse registration results are quite accurate even though it has been calculated from a single pair of matching tensors in each case and no registration refinement has been performed yet. (This figure is best viewed in colour).

$\mathbf{M'}_m$ and $\mathbf{M'}_s$ are registered to form $\mathbf{M'}_{ms}$. $\mathbf{M'}_{ms}$ is then aligned along its principal axis and its x, y, z bounding dimensions $\mathbf{D'}_{ms}$ are calculated using Eq. (9).

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

In Eq. (9), $\mathbf{V'}_{ms}$ is the matrix of x, y, 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, the bounding dimensions $\mathbf{D}$ of the object are subtracted from $\mathbf{D'}_{ms}$. If the maximum difference between the two is less than a specified tolerance t_D, $\mathbf{M}_m$ and $\mathbf{M}_s$ are also registered (using the same $\mathbf{R}$ and $\mathbf{t}$) and pairs of points on $\mathbf{M}_m$ and $\mathbf{M}_s$ that are within a distance equal to $2d_{\text{res}}$ (where d_{res} is the resolution of $\mathbf{M}_i$) are turned into correspondences. t_D is chosen as a fraction of $\mathbf{D}$. Since the registration at this stage is calculated from a single set of matching tensors,it is likely to be inaccurate. Therefore, a high value of t_D is selected. In our experiments we chose $t_D = \text{mean}(D_x, D_y, D_z)/10$. If the number of correspondences found is more than n_c ($n_c = \min(\text{number of points in } \mathrm{M}_m, \mathrm{M}_s)/4$), the registration is refined with a *variant* of the ICP algorithm (Rusinkiewicz and Levoy, 2001). Correspondences are established once again between points on the two views that are within a distance equal to d_{res}. If the number of correspondences found is more than $2n_c$, the combined bounding dimensions $\mathbf{D}_{ms}$ of the registered meshes ($\mathbf{M}_m$ and $\mathbf{M'}_s$) are calculated in a similar fashion using Eq. (9). Once again $\mathbf{D}$ is subtracted from $\mathbf{D}_{ms}$. If the maximum difference between the two bounding dimensions $\mathbf{D}_{ms}$ and $\mathbf{D}$ is less than a predefined tolerance $2d_{\text{res}}$, the transformation is accepted. A very low value of tolerance is selected at this stage compared to t_D since the registration has now been refined.If any one of the above `local-verification` steps fails, the next pair of tensors is selected for matching and the whole process is repeated.

4.2. *Global Verification (Steps 15 and 16 of the Pseudo-Code of Fig. 7)*

In case there are more than two views, they are all pairwise registered in the coordinate basis of a reference view as described above. Each time a new view is added to the set of registered views, `global-verification` is performed by calculating the combined bounding dimensions of all the views (Eq. (9)), that are registered so far, and comparing them with $\mathbf{D}$. If the maximum difference between the two is less than $4d_{\text{res}}$, the newly added view is accepted. If `global-verification` fails, the pairwise correspondence algorithm is repeated for the last pair of views and their next pair of tensors is matched. Pseudo code of our automatic correspondence and registration algorithm is given in Fig. 7.

Note that in our approach, automatic coarse registration is performedon the basis of a single pair of matching tensors. An alternate possibilityis to match a predetermined number of tensors and calculate the rigid transformation supported by the maximum number of matching tensors using RANSAC. This approach could be more robust but it will be computationally more expensive as there are six degrees of freedom when registering two range images. Moreover, the coarse registration resulting from a single pair of matching tensors is quite accurate (see Fig. 8) and serves as a reliable starting point for an onward refinement with a fine registration algorithm (e.g. ICP (Besl and McKay, 1992)).

5. The 3D Modeling Framework

We combined our automatic pairwise registration algorithm with other modular components to devise a complete framework of automatic 3D modeling from ordered range images i.e. with known overlapping view pairs.[2] Figure 9 shows the block diagram of our framework.The input to our framework is the ordered set of views in the form of point clouds. No other information is required by the algorithm and nor does the algorithm make any assumption about the viewing angles, overlapping regions or the shape of the object. The input views are pairwise registered using our automatic algorithm described in Section 4 and the registration is further refined with a global registration algorithm (Williams and Bennamoun, 2001) (Module 5 of Fig. 9) in order to distribute any registration errors evenly over the complete 3D model. Finally, the registered views are integrated and reconstructed using the Volumetric Range Image Processing Package (VripPack) (Stanford Computer Graphics Laboratory, 2001) as shown in Module 6 of Fig. 9. VripPack uses the volumetric integration algorithm by Curless and Levoy (1996) for integration and the marching cubes algorithm (Lorensen and Cline, 1987)

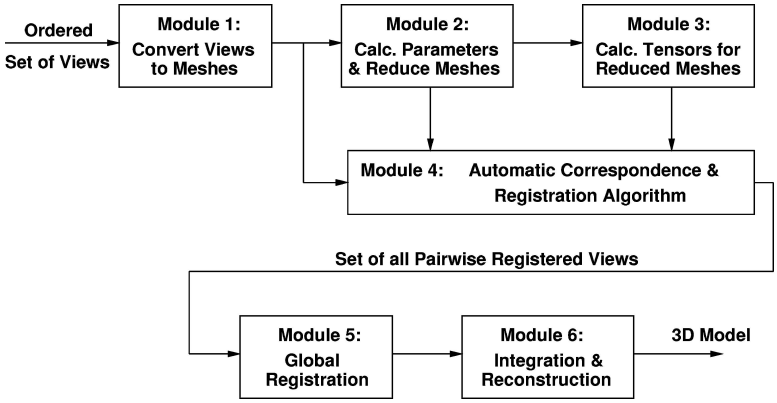

Figure 9. Block diagram of our automatic 3D modeling framework. (Figure reproduced from Mian et al. (2004d)).

for reconstruction. The global registration algorithm, integration algorithm and surface reconstruction algorithm are all modular components of our 3D modeling framework and can eventually be replaced with better algorithms.

6. Results

We performed our experiments on range data obtained from different sources on the Internet. Figure 10 shows our 3D modeling results using Johnson's data set (Johnson and Hebert, 1999; Johnson, 1997). Figure 11 shows our 3D modeling results using range data from The Stuttgart Range Image Database (The University of Stuttgart, 2001). We performed our experiments on range images of different types of free-form objects with varying properties of features, symmetries, planar regions, curvatures etc. in order to test our algorithm in every possible scenario. Since the ground truth was not available in all these cases, we could only perform a qualitative analysis of the pairwise registrations and their resulting 3D models. The registered views were magnified and visually inspected for alignment errors and seams. There were no visually noticeable defects or seams in the registered views. Moreover, the reconstructed models in Figs. 10 and 11 also look very accurate.

We also performed some experiments on range data for which the ground truth was available. These range images were synthetically generated from already built models available at The Stanford 3D Scanning Repository (Stanford Computer Graphics Laboratory, 2003) namely the armadillo, the Stanford bunny and the happy buddha. 26 synthetic range images were generated for each model (using a z-buffer) from different viewpoints 30° apart. These range images were then automatically registered with our algorithm

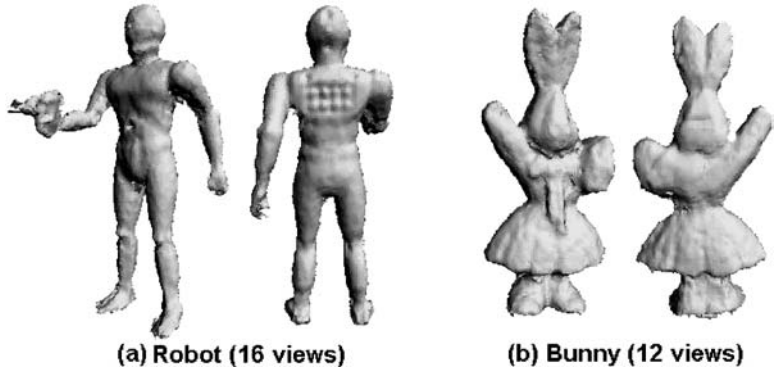

Figure 10. Our 3D modeling results using Johnson's range data. (a) 16 views of the robot were registered to construct its 3D model. (b) 12 views of the bunny were registered to construct its 3D model.

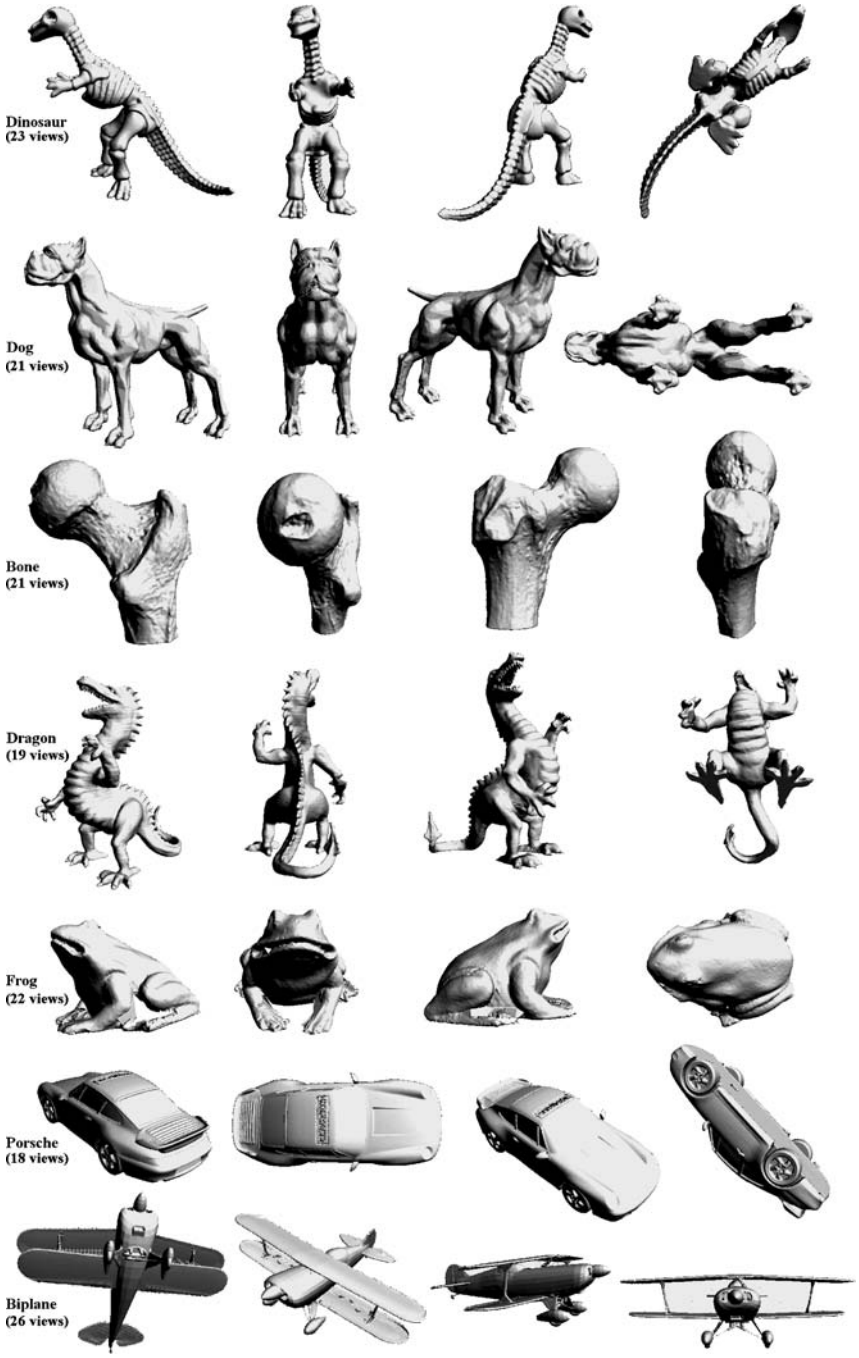

Figure 11. Our 3D modeling results using range data from the Stuttgart Range Image Database. Each 3D model is shown from four different angles. The number of views used to construct each model is written to its left below its name. Notice that these models look more neat compared to the models in Fig. 10 because the range data was of a better quality in this case.

and reconstructed with VripPack (Curless and Levoy, 1996). Figure 12 shows the original 3D models along with the rebuilt ones. Visual comparison of the models showed that the rebuilt models are identical to their original counterparts except for minute and indiscernible blurring which occurred in the rebuilt models due to approximations in the integration and reconstruction phases.

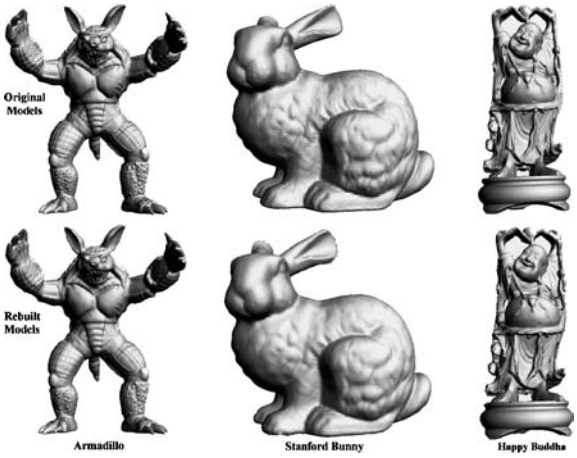

Figure 12. 3D modeling results of synthetic data. 26 different views of each model were synthetically generated from view points 30° apart. The models were rebuilt from these views using our framework. The rebuilt models (second row) are identical to the original models (first row) except for minute and indiscernible blurring.

7. Quantitative Analysis of the Automatic Registration Algorithm

In addition to the qualitative analysis of our 3D models, we also performed extensive testing of our automatic pairwise correspondence and registration algorithm. The algorithm was tested according to the following criteria: (1) Accuracy (2) Robustness to resolution and the number of tensors perview (3) Efficiency with respect to memory and time (4) Required amount of overlap and (5) Robustness to noise. Details are given below with respect to each criterion.

7.1. Accuracy

To analyze the accuracy of our algorithm quantitatively, it was necessary to generate range images with available ground truth transformations. This test was therefore performed on the synthetically generated range images of the Stanford 3D models (Stanford Computer Graphics Laboratory, 2003).While breaking these 3D models into 26 views, the ground truth rotation matrix ($\mathbf{R}_{iGT}$) and translation vector ($\mathbf{t}_{iGT}$) with respect to a reference view were recorded for each view $\mathbf{V}_i$ ($i = 1, 2 \ldots, 26$). Next, the transformations ($\mathbf{R}_i$) and $\mathbf{t}_i$ resulting from our automatic registration algorithm were compared to the ground truth transformations. The error in the two rotation matrices was calculated using Eqs. (10) and (11).

$$\mathbf{R}_{id} = \mathbf{R}_i \mathbf{R}_{iGT}^{-1} \tag{10}$$

$$\theta_{ie} = \cos^{-1}\left(\frac{\text{trace}(\mathbf{R}_{id}) - 1}{2}\right)\frac{180}{\pi} \tag{11}$$

In Eq. (10), $\mathbf{R}_{id}$ is a rotation matrix representing the difference between $\mathbf{R}_i$ and $\mathbf{R}_{iGT}$. $\mathbf{R}_{id}$ is equal to an identity matrix in case of no error. Equation (11) is derived from Rodrigue's formula. θ_{ie} represents the amount of rotation error (about a single axis) present in $\mathbf{R}_i$. Similarly, the translation error t_{ie} of each view $\mathbf{V}_i$ was calculated using Eq. (12).

$$t_{id} = \frac{||\mathbf{t}_i - \mathbf{t}_{iGT}||}{d_{\text{res}}} \tag{12}$$

In Eq. (12), d_{res} is the resolution of the fine meshes i.e. before mesh reduction. The difference between the translation vectors is normalized with respect to d_{res} in order to make it scale-independent. Figure 13 shows histograms of the errors in rotation and translation of the 26 views of all 3D models. Most of the pairs have a rotation error less than 0.1° and a translation error less than 0.1 mesh resolution.

7.2. Robustness to Resolution and the Number of Tensors per View

We tested our algorithm's performance by varying the resolution of the range images and the number of tensors per view. We varied the resolution of the views of the objects of Fig. 11 by simplifying (Garland and Heckbert, 1997) them by different factors. Figure 14 shows some example views at the lowest three resolutions. Next, we used our automatic algorithm for pairwise registration of overlapping views and categorized the results as correct or incorrect. Figure 15 shows the number of overlapping view pairs which were correctly or incorrectly registered at different resolution for six individual objects. Note that these results are reported for individual pairs ofoverlapping views without performing `global-verification`. Many of these incorrect registrations were detected at the `global-verification` stage and hence corrected by searching for another pair of matching tensors which satisfies the `global-verification` stage. Figure 16 shows the combined results of all the seven objects at varying resolution and number of tensors per view. We can see in Fig. 16 that

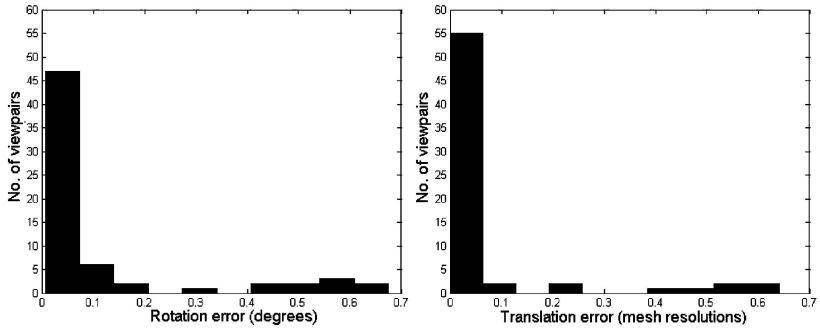

Figure 13. Histograms of errors in rotation and translation after registration (using our algorithm) of the synthetically generated views of the Stanford bunny, armadillo and happy buddha. Maximum view pairs have a rotation error less than 0.1° and a translation error less than 0.1° mesh resolution. The translation error is normalized with the mesh resolution (before simplification) to make it scale independent. The average dimension i.e. the average of height, length and width, of each object are as follows. Stanford bunny: dimension = 150 cm resolution = 1.47 cm, armadillo: dimension = 125 cm resolution = 0.53 cm and happy buddha: dimension = 121 cm and resolution = 0.72 cm. (Figure reproduced from Mian et al. (2004d).)

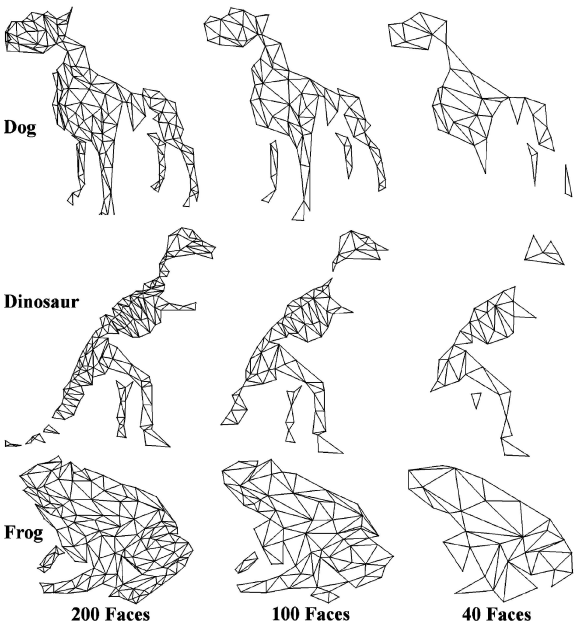

Figure 14. Example views of the dog, the dinosaur and the frog at their lowest three resolutions. Notice that the objects start loosing their meaningful shape at 100 faces per view.

80% correct results are acheived at resolutions as low as 200 faces per view and 175 points per view. Note that the objects lose their meaningful shape at a resolution of 100 faces, yet we still achieved 50% correct pairwise registrations at this resolution (see Fig. 16). The performance of our algorithm with varying number of tensors (Fig. 16 last column) shows that our algorithm achieves a success rate of approximately 90% at 450 tensors per view. With `global-verification` a success rate of almost 100% was acheived at 400 faces per view and 500 tensors per view. Note that these figures are independent of the size of the actual range images i.e. whatever is the original resolution of the range images, our algorithm reduces them to 400 faces per view and represents each view by 500 tensors to achieve correct results.

7.3. *Efficiency with Respect to Memory and Time*

Our algorithm is efficient in terms of memory utilization because a limited and constant number of tensors (approx. 500) are required to represent each view. Furthermore, as explained in Section 2, we take advantage of the sparsity of these tensors and compress them to further cut down on memory utilization. Computational efficiency is achieved by performing coarse registration (matching) at a very low resolution and by matching only a small number of tensors. Our experiments show that most of the time a correct pair of matching tensors is found when the first few tensors of $\mathbf{M}'_s$ are matched with the tensors of $\mathbf{M}'_m$. Figure 17(a) shows a histogram of the number of tensors of $\mathbf{M}'_s$ that were matched with the tensors of $\mathbf{M}'_m$ during the experiments of Section 7.2. Most of the correct matches were found when the first 50 tensors of any $\mathbf{M}'_s$ were matched with those of $\mathbf{M}'_m$. The median number of tensors of $\mathbf{M}'_s$ that were matched[3] with the tensors of $\mathbf{M}'_m$ was found to be 26. Once a correct match which passes the `local-verification+` and `global-verification` is found, the views are registered and our algorithm stops searching for any further matches consequently saving computational time.

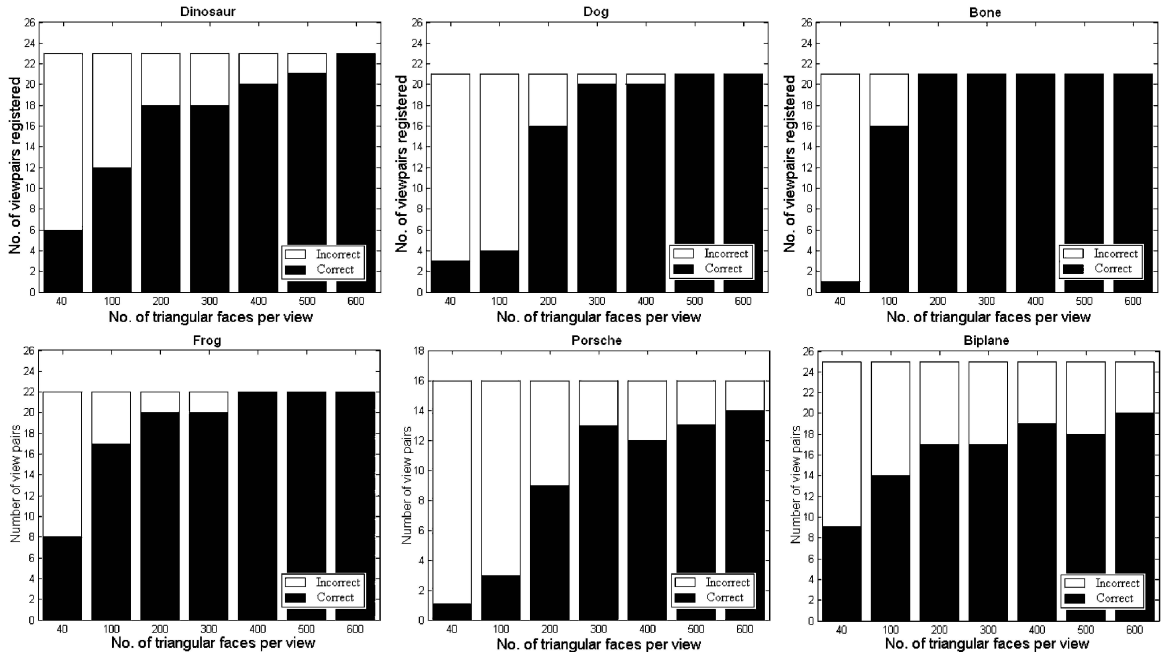

Figure 15. Performance of our automatic registration algorithm with varying mesh resolution. Individual results for six objects. The biplane has the maximum number of incorrect matches due to its highly symmetrical shape. Note that these results have been recorded for individual pairs of overlapping views only and these incorrect matches were detected and corrected during `global-verification`.

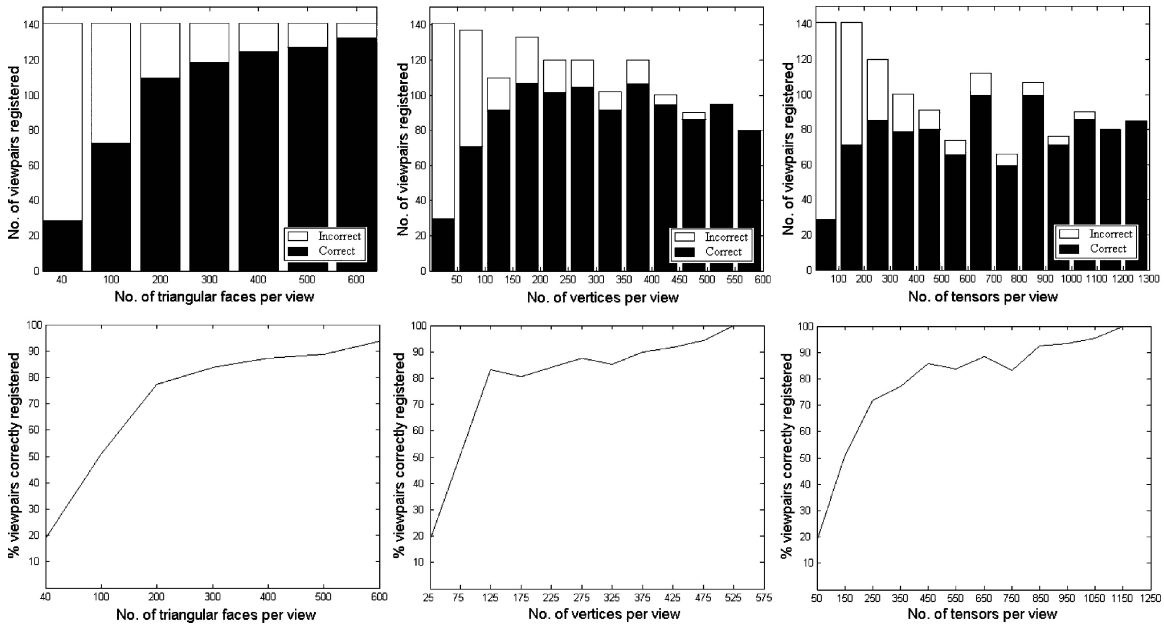

Figure 16. Performance of our automatic registration algorithm with varying mesh resolution. Combined results for all seven objects. The algorithm performs well even at very low resolutions. Note that these results (Figs. 15 and 16) are recorded for pairs of views only and these incorrect matches were detected and corrected during `global-verification`. (Figure reproduced from Mian et al. (2004d).)

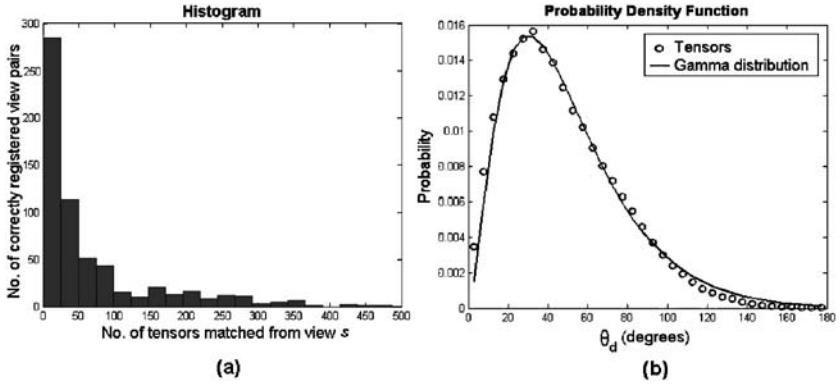

Figure 17. (a) A histogram of the number of tensors of $\mathbf{M}'_s$ that were matched with $\mathbf{M}'_m$. Most of the correct correspondences were found when the first 50 tensors of $\mathbf{M}'_s$ were matched. (b) A PDF of θ_d of the tensors of all the views of the 7 objects of Fig. 11. (See text for details.)

We will now calculate the worst case complexity of our tensor matching algorithm i.e. when all the tensors of one view are matched with all the tensors of another view. The complexity of any one-to-one matching algorithm is $O(n_t^2)$ (where n_t is the total number of descriptors per view). However, the number of tensors per view in our case is constant i.e. n_t is constant. This means that the complexity of our automatic coarse registration algorithm is independent of the size of the range images. Moreover, the number of tensors that are actually matched in our case is much less than n_t^2 because of the use of indexing. The improvement factor $\frac{n_t^2}{\mu}$ (where μ is the number of tensors matched when indexing is used) depends upon the probability density of θ_d of the tensors. Figure 17(b) shows the probability density function (PDF) of the θ_d of the tensors of all the views of the seven objects of Fig. 11. This PDF closely follows a Gamma distribution with $\alpha = 2.5$ and $\beta = 20$. The more flat this PDF is, the greater the improvement factor is and vice versa. Suppose there are 500 tensors per view ($n_t = 500$) and each tensor of the first view is matched with only those tensors of the second view which are indexed by $\theta_d \pm \Delta\theta_d$. The total number of tensors (μ) that are matched in the worst case is given by Eq. (13).

$$\mu = 3n_t^2 \sum_{i=1}^{180/\Delta\theta_d} \left(\int_{\theta_{d_i}}^{\theta_{d_i}+\Delta\theta_d} \Gamma(\theta_d) d\theta_d \right)^2 \approx 40432 \tag{13}$$

In Eq. (13), $\Gamma(\theta_d)$ is the gamma PDF. The PDF is multiplied by 3 because each tensor is matched with the tensors indexed by $\theta_d \pm \Delta\theta_d$ (this range of θ_d will point to 3 bins in the index table). $\mu \approx 40432$ when $n_t = 500$ and $\Delta\theta_d = 5^\circ$. Without the use of an index table, this figure would have been $(500)^2 = 250000$. The improvement factor in this case is $\frac{250000}{40432} = 6.2$. Therefore, in the worst case our algorithm would execute six times faster when using indexing than it would execute without the use of an index table. A Matlab implementation of our algorithm on a 2.4 GHz machine with 512 MB memory takes 15 seconds to pairwise register two overlapping range images in the worst case i.e. when all tensors of view s are matched with all the tensors of view m using the index table. However, as Fig. 17(a) shows, in most cases our algorithm finds a correct registration when the first few tensors of view s are matched with those of view m. Moreover, the execution time is expected to improve many folds once our algorithm is implemented in C++.

7.4. *Required Amount of Overlap*

In this experiment, we tested the performance of our algorithm against varying amounts of overlap between the range images to be registered. These experiments were performed on the range images of the dinosaur, the dog, the bone and the biplane. We define the amount of overlap between two meshes $\mathbf{M}_m$ and $\mathbf{M}_s$ according to Eq. (14). For each of the four objects, the overlap was calculated between all possible $N(N-1)/2$ pairs of views[4] (N is the total number of views per object). Next, we used our automatic algorithm for pairwise registration of each of the $N(N-1)/2$ view pairs and categorized the results as correct or incorrect. Figure 18 shows the results of our experiments. There is some variation between the results of individual objects but

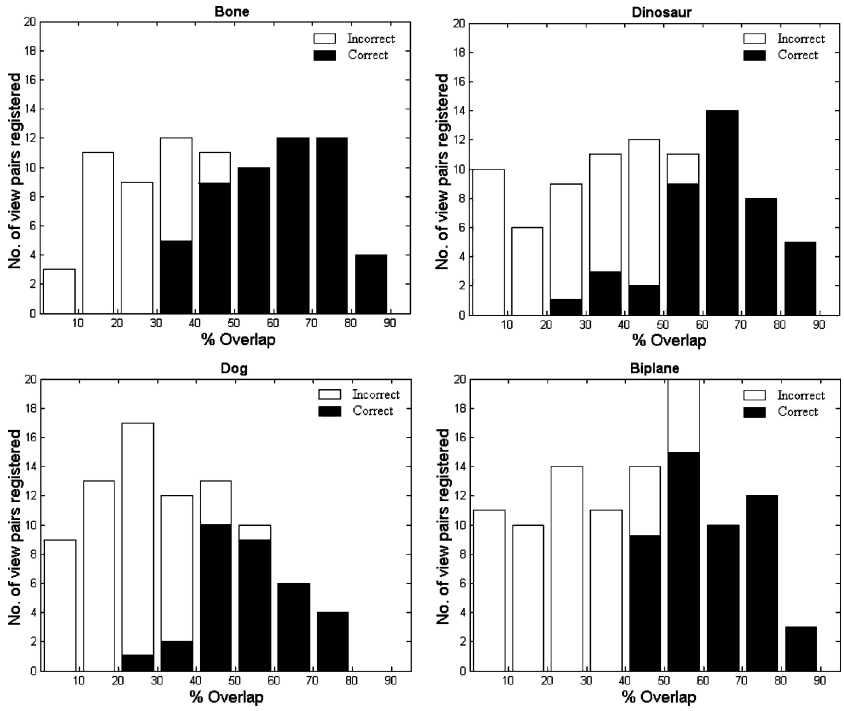

Figure 18. Performance of our algorithm as a function of the amount of overlap. The results vary with the type of object. A 50% and above overlap however ensures correct matches independently of the object. (Figure reproduced from Mian et al., 2004d.)

generally an overlap of 50% or more ensures a correct match.

$$\text{Overlap} = \frac{\text{no. of corresponding verties of } \mathbf{M}_m \text{ and } \mathbf{M}_s}{\min(\text{no. of vertices in } \mathbf{M}_m,\ \text{no. of vertices in } \mathbf{M}_s)} \quad (14)$$

7.5. *Robustness to Noise*

We used the range images of the dinosaur, the dog and the bone for this test. Gaussian noise with standard deviation $\sigma = d_{\text{res}}, 2d_{\text{res}}, 3d_{\text{res}}, 4d_{\text{res}}$ (where d_{res} is the resolution of the range image) was injected into these range images along the scanner viewing direction.[5] Next, our algorithm was used to automatically register overlapping view pairs. Close to 100% correct registrations were achieved up to $\sigma = 3d_{\text{res}}$ whereas some of the view pairs could not be registered at $\sigma = 4d_{\text{res}}$. The robustness of our algorithm to noise can be attributed to two main reasons. First, our algorithm has a mesh reduction phase at the beginning which reduces the effect of noise. Notice that despite the spiky surface of the views of the dog (Fig. 19, column 2), the output of the mesh reduction algorithm (Garland and Heckbert, 1997) is quite smooth (Fig. 19, column 3). Second, our algorithm uses a correlation coefficient for matching tensors. Correlation coefficient, being a statistical measure, performs better in the presence of noise as compared to other matching techniques (for example linear matching). The first column of Fig. 19 shows two views of the dog. Noise with $\sigma = 4d_{\text{res}} = 2.8$ cm has been added to these views in the second column. In column three, the noisy meshes are reduced and finally column four shows the result of their registration. We can see that the registration is correct even though most of the features on each view of the dog were distorted due to the addition of noise.

8. Comparison with Spin Images

A major criterion while developing our representation was to achieve higher discriminating capability than existing representations. Such a representation will result in comparatively more accurate matches. Our representation calculates third order tensors over the 3D surface which describe the underlying local surface patches more distinctly compared to other

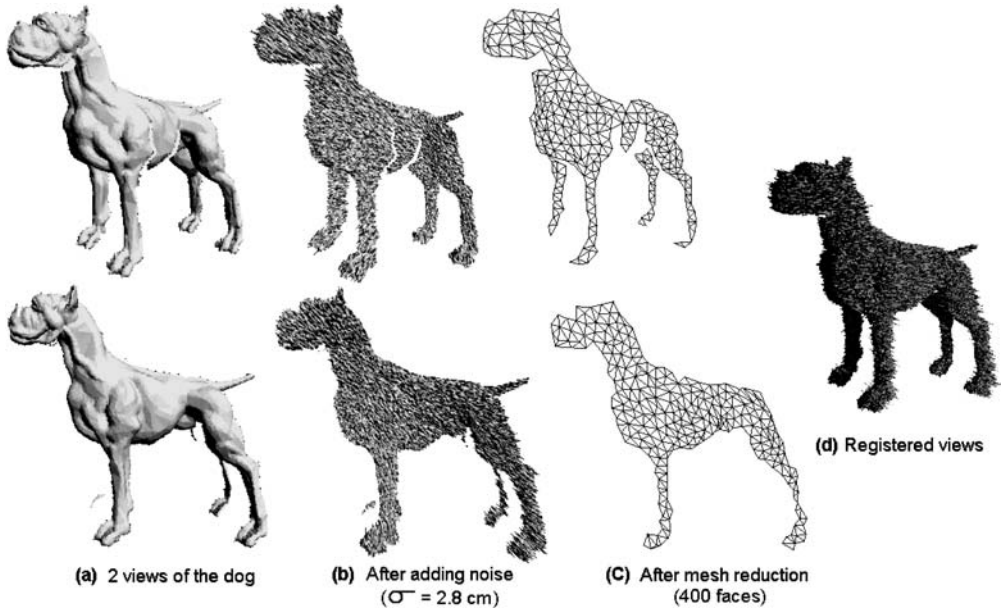

Figure 19. (a) Two views of the dog. (b) After adding Gaussian noise with $\sigma = 4d_{\text{res}} = 2.8$ cm, most of the features of the dog are distorted. (c) Despite the spiky surface of the noisy views, the output of the mesh reduction is smooth. (d) The noisy views are correctly registered by our algorithm.

representations which map the 3D surface onto a 2D histogram (e.g. spin images (Johnson and Hebert, 1997) and geometric histograms (Ashbrook et al., 1998)) or the ones which extract 1D signatures from the range image (e.g. point signatures (Chua and Jarvis, 1997)). Because of the higher discriminating capability of tensors, our algorithm can register meshes at a very low resolution (see Section 7.2).

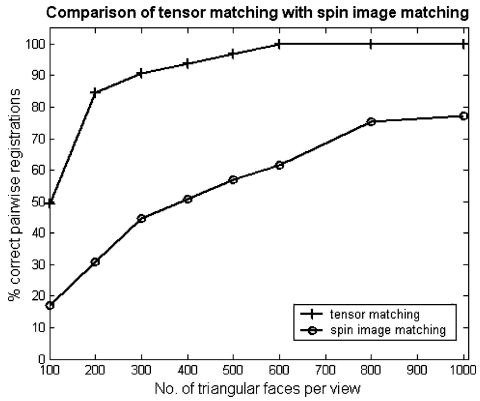

Figure 20. Comparison of our tensor matching algorithm with the spin image matching algorithm. Tensor matching performed better than the spin images. Note: the number of bins of a tensor were $10\times10\times10$ where as the number of bins of a spin image were 15×15.

We compared our algorithm with the spin image matching algorithm (code available at (Mesh Tool Box, 2004)) by applying both algorithms to the same data set i.e. the views of the dinosaur, the dog and the bone. This amounted to a total number of 65 pairwise registration tests for each algorithm. The tests were performed to compare the performance of the two algorithms at varying resolution of the views. Figure 20 shows our results. Our algorithm performed much better than the spin images by achieving 100% correct pairwise registrations at 600 faces per view whereas only 61% of the views were correctly registered by the spin images at this resolution. At 200 faces per view, our algorithm's performance dropped to 84% (16% decrease) whereas the peformance of the spin images dropped to 30% (50% decrease). This shows that our representation is more robust to the resolution of the views compared to the spin images.

In another experiment we compared the discriminating capability of our tensor representation to the spin images. For better control over the spin image generation parameters and the matching criterion, we used our own implementation of the spin images for this experiment. Moreover, we used the same data used by Johnson (1997) i.e. two views of the robot (see Fig. 22), to ensure that the data suits the spin image

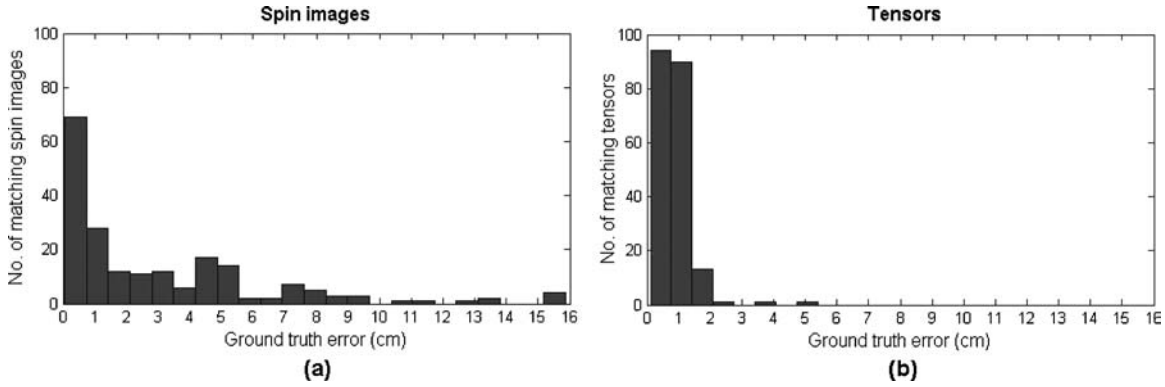

Figure 21. Histograms of the ground truth error between the matching pairs of (a) spin images and (b) tensors. 97.5% of the matching pairs of tensors have less than 2 cm error whereas only 53.5% of the matching pairs of spin images have less than 2 cm error. This proves that our representation is more discriminating and gives more accurate correspondences compared to the spin images.

matching algorithm. The parameters of the tensors and the spin images were closely matched. The bin size was selected equal to the mesh resolution i.e. 0.41 cm in each case. The number of bins of the spin images were 15×15 whereas the number of bins of the tensor were $15 \times 15 \times 15$. The number of spin images and the number of tensors per view was also fixed to 700. Next, the tensors and spin images of the two views were matched using a similar criterion i.e. a linear correlation coefficient. The best 200 matches, based on their correlation coefficient, were recorded in each case. Next, the ground truth[6] displacement error between the matching pairs of descriptors was calculated. Figure 21 shows the histograms of these errors for the two descriptors. 97.5% of the matching pairs of tensors have an error of less than 2 cm whereas only 53.5% of the matching pairs of spin images have an error of less than 2 cm. These results clearly indicate that our tensor representation is more discriminating resulting in a higher number of accurate matches compared to the spin image representation.

Note also that a tensor calculated with the above parameters is more local compared to a spin image with equal parameters. This is because a spin image of 15×15 bins sweeps exactly π times more volume than the volume enclosed by a tensor of $15 \times 15 \times 15$ bins and of equal bin size (see caption of Fig. 22). This fact is illustrated pictorially in Fig. 22. Even though a tensor is more local i.e. describes a smaller surface, it has more discriminating power compared to the spin images (see Fig. 21). Since a tensor is a more local representation, it also gives better results when used for recognition of occluded objects (Mian et al., 2004a). The reader may also note that all the previous experiments including

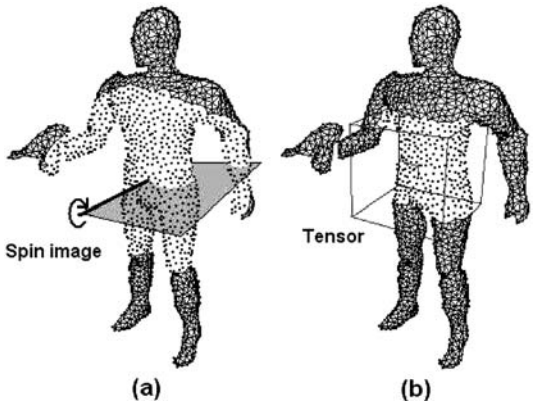

Figure 22. (a) A spin image is generated by an image plane spinning about the normal of a vertex (shown by a thick line) and summing vertices (shown distinctly in the figure without triangulation) as they pass through the bins of the image plane. A 15×15 bins spin image with unit bin size therefore sweeps a cylindrical volume of $\pi r^2 h = \pi 15^3$ (where r is the width and h is the height of the image plane; in this case $r = h$). (b) A $15 \times 15 \times 15$ bins tensor with unit bin size encloses a volume of 15^3. The vertices enclosed by the tensor are shown without triangulation for comparison with the spin image of (a). The cylindrical volume swept by a spin image (a) is exactly π times greater than the volume enclosed by a tensor (b) of equal parameters. Therefore, with equal parameters, a tensor is a comparatively more local descriptor with greater discriminating power (see Fig. 21).

those reported in Fig. 21 were performed with tensors of $10 \times 10 \times 10$ bins. Another strength of our approach is that a single pair of matching tensors gives a unique transformation that aligns the two views (Eqs. (7) and (8)). Moreover, because of the higher dimensionality (x, y, z, θ_d) of the tensors, there is more potential for multidimensional indexing which can further speed up the matching process. For example, our tensors can be

used in conjunction with a 4D hash table for simultaneous matching of a single tensor with multiple tensors (Mian et al., 2004a).

Although the 2D spin image representation is less descriptive compared to its 3D tensor counterpart, it has a positive side that it relies on a single vertex and its normal. Moreover, a spin image taken in *isolation*, requires less memory compared to a tensor of equal parameters. This makes the correlation between a pair of spin images comparatively faster. However, its limited descriptiveness necessitates the computation and subsequent correlations between a larger number of descriptors (i.e. spin images) compared to the tensor representation. Furthermore, any gain in speed is diminished by the high computational cost requiredto prune the large number of incorrect matches attributed to the low descriptiveness of the spin images.

9. Conclusion

In this paper, we presented a novel 3D free-form object representation scheme based on third order tensors. Our tensor representation has more discriminating capability which results in accurate pairwise correspondences. We also presented a fully automatic correspondence and registration algorithm by efficiently matching tensors of overlapping views. The automatic registration algorithm was integrated with other modular components to form a complete framework of 3D modeling which automatically generates a 3D model from the range images of an object with known overlapping view pairs. Our automatic registration algorithm is applicable to free-form objects and does not require any knowledge of the viewing angles or the regions of overlap of the views. We performed our experiments on range images obtained from different sources and presented our 3D modeling results. We also performed extensive testing of our automatic pairwise registration algorithm according to an number of important criteria. Our results show that our algorithm is accurate and efficient. Moreover, it is robust to resolution, number of tensors per view, the required amount of overlap and noise. Our tensor representation gives better results compared to the spin image matching algorithm when applied on the same range images.

Our tensor representation is general and can be extended to represent more features than just the 3D surface area of an object. The use of surface normals in addition to area for generating fourth order tensors has been demonstrated in (Mian et al. (2004a). Extending our representation to include even more features will further enhance its discriminating capability leading to robust 3D object recognition. In the future, we plan to use the tensor representation for automatic multiview coarse registration of unordered range images when a priori knowledge of the overlapping view pairs is not available.

Acknowledgments

The authors would like to acknowledge the following institutions. Carnegie Mellon University, USA for providing range data and mesh toolbox; Stanford University, USA for providing 3D models and VripPack; The University of Stuttgart, Germany for providing range data. This research is sponsored by ARC grant number DP0344338.

Notes

1. This may appear to be a low percentage however it is important to note that even the vertices which are outside the overlapping region have been taken into consideration when calculating this percentage. This is also the amount of overlap between the two meshes calculated using Eq. (14).
2. This information can easily be extracted from the order of the scans.
3. 26 tensors of $\mathbf{M}'_s$ were matched with only those tensors of $\mathbf{M}'_m$ which were indexed by $\theta_d \pm \Delta\theta_d$ in the index table.
4. The overlap was calculated after the views were registered using transformations calculated from our earlier experiments.
5. Noise in range images generally occurs in the scanner viewing direction.
6. Since ground truth was not available, manual coarse registration and refinement with the ICP algorithm was considered as the ground truth.

References

Ashbrook, A.P., Fisher, R.B., Robertson, C., and Werghi, N. 1998. Finding surface correspondence for object recognition and registration using pairwise geometric histograms. *International Journal of Pattern Recognition and Artificial Intelligence*, 2:674–686.

Benjemma, R. and Schmitt, F. 1997. Fast global registration of 3D sampled surfaces using a multi-Z-buffer technique. In *International Conference on Recent Advances in 3D Digital Imaging*, pp. 113–120.

Besl, P. 1990. *Machine Vision for Three-dimensional Scenes*. Academic Press.

Besl, P.J. and McKay, N.D. 1992. Reconstruction of real-world objects via simultaneous registration and robust combination of

multiple range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Campbell, R.J. and Flynn, P.J. 2001. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210.

Chen, C., Hung, Y., and Cheng, J. 1991. RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1229–1234.

Chen, Y. and Medioni, G. 1991. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation*, pp. 2724–2729.

Cheng, J. and Don, H. 1991. A graph matching approach to 3-D point correspondences. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(3):399–412.

Chua, C.S. and Jarvis, R. 1996. 3D free-form surface registration and object recognition. *International Journal of Computer Vision*, 17:77–99.

Chua, C.S. and Jarvis, R. 1997. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85.

Curless, B. and Levoy, M. 1996. A volumetric method for building complex models from range images. In *Computer Graphics, SIGGRAPH*.

Foley, J., van Dam, A., Feiner, S.K., and Hughes, J.F. 1990. *Computer Graphics-Principles and Practice*, 2nd ed. Addison-Wesley.

Garland, M. 1999. Quadric-based polygonal surface simplification. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.

Garland, M. and Heckbert, P.S. 1997. Surface simplification using quadric error metrics. In *SIGGRAPH*, pp. 209–216.

Higuchi, K., Hebert, M., and Ikeuchi, K. 1994. Building 3-D models from unregistered range images. In *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2248–2253.

Johnson, A.E. 1997. Spin images: A representation for 3-D surface matching. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Available along with range data at http://robotics.jpl.nasa.gov/people/johnson/thesis/thesis.html.

Johnson, A.E. and Hebert, M. 1997. Surface registration by matching oriented points. In *International Conference on Recent Advances in 3-D Imaging and Modelling*, pp. 121–128.

Johnson, A.E. and Hebert, M. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):674–686.

Lorensen, W.E. and Cline, H.E. 1987. A high resolution 3D surface construction algorithm. In *Computer Graphics. ACM SIGGRAPH*, pp. 163–169.

Mesh Tool Box, 2004. Vision and mobile robotics laboratory, Carnegie Mellon University, available at http://www-2.cs.cmu.edu/vmr/software/meshtoolbox/downloads.html.

Mian, A.S., Bennamoun, M., and Owens, R.A. 2004a. A novel algorithm for automatic 3D model-based free-form object recognition. In *IEEE International Conference on Systems, Man and Cybernetics*, pp. 6348–6353.

Mian, A.S., Bennamoun, M., and Owens, R.A. 2004b. From unordered range images to 3D models: A fully automatic multiview correspondence algorithm. In *Theory and Practice of Computer Graphics*. IEEE Computer Society Press, pp. 162–166.

Mian, A.S., Bennamoun, M., and Owens, R.A. 2004c. Matching tensors for automatic correspondence and registration. In *European Conference on Computer Vision*, vol. 2, pp. 495–505.

Mian, A.S., Bennamoun, M., and Owens, R.A., 2004d. Performance analysis of an improved tensor based correspondence algorithm for automaic 3D modeling. In *IEEE International Conference on Image Processing*, pp. 1951–1954.

Nishino, K. and Ikeuchi, K. 2002. Robust simultaneous registration of multiple range images. In *Asian Conference on Computer Vision*, pp. 454–461.

Oishi, T., Sagawa, R., Nakazawa, A., Kurazume, R., and Ikeuchi, K. 2003. Parallel alignment of a large number of range images. In *International Conference on 3-D Digital Imaging and Modeling*, pp. 195–202.

Rangarajan, A., Chui, H., and Duncan, J. 1999. Rigid point feature registration using mutual information. *Medical Image Analysis*, 3(4):425–440.

Roth, G. 1999. Registering two overlapping range images. In *IEEE International Conference on 3-D Digital Imaging and Modeling*, pp. 191–200.

Rusinkiewicz, S. and Levoy, M. 2001. Efficient variants of the ICP algorithm. In *3DIM*, pp. 145–152.

Stanford Computer Graphics Laboratory, 2001. A volumetric range image processing package. http://graphics.stanford.edu/software/vrip/.

Stanford Computer Graphics Laboratory, 2003. The Stanford 3D scanning repository. http://graphics.stanford.edu/data/3Dscanrep/.

Stephens, R.S. 1990. A probabilistic approach to the hough transform. In *British Machine Vision Conference*, pp. 55–59.

The University of Stuttgart, 2001. Stuttgart range image database. http://range.informatik.uni-stuttgart.de/htdocs/html/.

Williams, J. and Bennamoun, M. 2001. Simultaneous registration of multiple corresponding point sets. *Computer Vision and Image Understanding*, 81(1):117–142.

Wyngaerd, J.V., Gool, L.V., Koth, R., and Proesmans, M. 1999. Invariant-based registration of surface patches. In *IEEE International Conference on Computer Vision*, vol. 1, pp. 301–306.