



UWA Research Publication

Mian, A., Bennamoun, M., & Owens, R. (2005). Automatic Correspondence for 3D Modelling: An Extensive Review. *International Journal of Shape Modeling*, 11(2), 253-291. 10.1142/S0218654305000797

© 2005 World Scientific Publishing Co Pte Ltd

This is pre-copy-editing, author-produced version of an article accepted for publication, following peer review. The definitive published version is located at <http://dx.doi.org/10.1142/S0218654305000797>

This version was made available in the UWA Research Repository on 4 March 2015, in compliance with the publisher's policies on archiving in institutional repositories.

Use of the article is subject to copyright law.

International Journal of Shape Modeling
© World Scientific Publishing Company

AUTOMATIC CORRESPONDENCE FOR 3D MODELING: AN EXTENSIVE REVIEW

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, bennamou, robyn}@csse.uwa.edu.au
<http://www.cs.uwa.edu.au>*

Received (Day Month Year)

Revised (Day Month Year)

Accepted (Day Month Year)

Communicated by (xxxxxxxxxx)

To generate a 3D computer model of a free-form object, multiple range images (point clouds) covering its entire surface are acquired from different viewpoints. These views are then aligned in a common coordinate basis by minimizing the distance error between their corresponding points. Establishing correspondences automatically is an inherently challenging problem due to the lack of any type of information other than the geometrical information extracted from the point clouds. Existing “automatic” correspondence techniques achieve automatism at the expense of other important specifications namely, applicability to free-form objects, accuracy, efficiency, robustness to resolution and surface sampling, robustness to overlap, robustness to noise and finally their applicability to simultaneous multiview correspondence. There is also a lack of a review paper that describes and critically analyzes these techniques. In this paper, we present such an extensive review and carry out the analysis of each technique according to the above listed indispensable criteria. Our analysis shows that none of these techniques fully meets these criteria and that there is still a need for the development of practical automatic correspondence algorithms.

Keywords: 3D modeling; correspondence; registration; free-form object; range image; 3D shape representation.

1. Introduction

There are numerous applications of three dimensional modeling in a wide range of areas. These areas include medical diagnosis, reverse engineering, robotics, computer graphics, computer games, preserving archaeological heritage, virtual reality and the entertainment industry. The first step in generating a 3D computer model of a free-form object³ is the acquisition phase during which the object’s surface is scanned from various viewpoints with a range scanner (see the block diagram in Fig. 1). This process results in a series of range images (also known as 2.5D images)

2 *Ajmal Mian et al.*

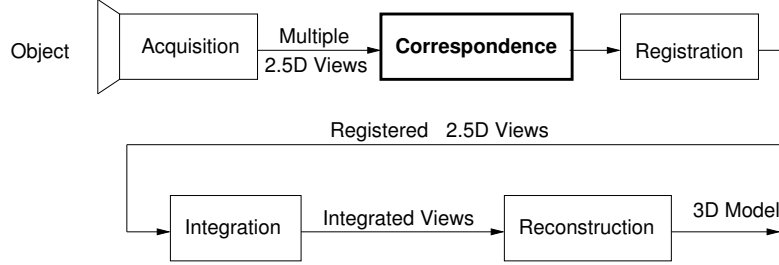


Fig. 1. A block diagram showing different components of the 3D modeling process.

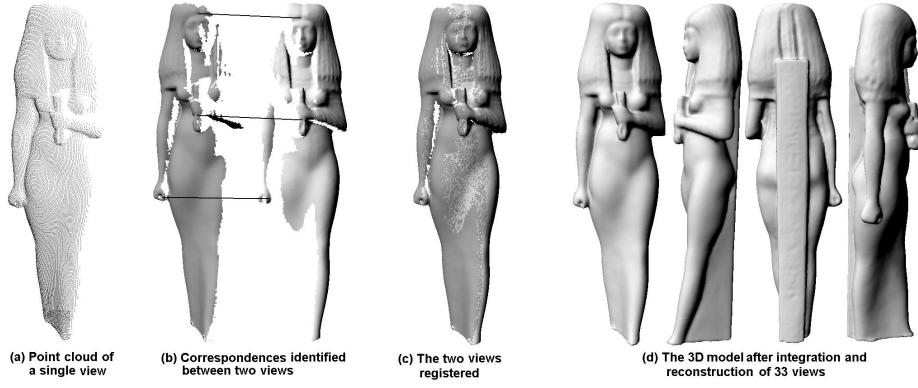


Fig. 2. (a) A point cloud of a view of the isis. (b) Three selected correspondences are marked between two views of the isis. (c) The two views after registration. Notice that the contribution of each view appears in a different gray shading. (d) The complete 3D model of the isis after integration and reconstruction of its 33 registered views.

of the object's surface in the form of point clouds (Fig. 2(a)). In the next step, correspondences are established between overlapping views. Points on two different views that correspond to the same point on the object are said to be corresponding points (Fig. 2(b)). These correspondences can then be used to derive a rigid transformation (rotation matrix \mathbf{R} and translation vector \mathbf{t}) that aligns the two views (see Fig. 2(c)). \mathbf{R} and \mathbf{t} are estimated with the objective of minimizing the distance error E between the corresponding pairs of points of the scene (\mathbf{S}) and the model (\mathbf{M}) based on Eqn. 1.

$$E = \sum_{i=1}^n \|\mathbf{S}_i \mathbf{R} + \mathbf{t} - \mathbf{M}_i\| \quad (1)$$

This is a common objective used by most existing registration algorithms however they use different strategies to establish correspondences between the views. Once all the views are registered in a single coordinate basis, they are integrated and reconstructed to make a complete seamless 3D model (Fig. 2(d)). Fig. 1 shows

the different components of the 3D modeling process and Fig. 2 illustrates the main phases of the 3D modeling process.

One way around the correspondence problem is to track the movement of the acquisition system and apply the reverse transformations to the views. However, this makes the acquisition device expensive and may also restrict its freedom of movement. Moreover, the accuracy of the registration is limited by the accuracy of the tracking system used. Therefore, a more economical and practical approach is to keep sufficient overlap between adjacent views and identify corresponding points in their regions of overlap as per the diagram in Fig. 1.

All the components of the 3D modeling process of Fig. 1, excluding the correspondence module, have already been automated and have reached a level of maturity. Examples of automatic acquisition techniques include and are not restricted to Ref. 5,35,46. For pairwise and simultaneous multiview registration, we cite Ref. 2,4,7,34,37,43. Examples of automatic integration techniques include Ref. 12,16,41 and finally, examples of automatic reconstruction techniques include Ref. 15,21,22,29.

However, automatic correspondence is an ongoing research problem and inherently challenging for the following reasons. First, each view of the object is defined in a separate coordinate frame and there is no available information related to the rotation and translation between the two coordinate frames. Consequently, the views cannot be directly matched in order to establish correspondences between them. Second, given that two views overlap, no information is available regarding their region of overlap. Therefore, it cannot be ascertained as to which points in one view have their corresponding points in the other view. Finally, in a more general case, given a set of unordered views of an object, there is no information regarding overlapping view pairs in that set.

In order to devise a fully automatic 3D modeling system, there is a need to develop an automatic correspondence algorithm which meets certain crucial specifications as listed below. We define an automatic 3D modeling system as one which is capable of generating a 3D model of an object from its *unordered* views without any user intervention. By unordered views, we mean that the order in which the views were acquired is unknown and hence there is no a priori knowledge about which view pairs overlap. The relative importance of these specifications depends upon the type of application. However, these specifications must be fully satisfied by a correspondence algorithm in order to be applicable in all practical scenarios. These specifications are:

- Automatism of the whole pairwise correspondence process
- Applicability to free-form objects
- Accuracy
- Efficiency with respect to time
- Robustness to the range image resolution and surface sampling
- Robustness to the overlap between adjacent views

- Robustness to the noise in the range images
- Applicability to multiview correspondence between unordered views

We define an *automatic correspondence* algorithm as one which requires no human intervention at any stage. A *free-form* object (also known as a sculpted object) is defined as one which has a smooth surface with a well defined and continuous surface normal everywhere except at vertices, edges and cusps³. No restrictions are imposed on the shape of the object. *Accuracy* of correspondences is necessary for accurate registration and 3D modeling. *Efficiency* with respect to time requires that the algorithm must execute in a reasonable duration of time. *Robustness to range image resolution* and *surface sampling* implies that an algorithm's computational complexity should not increase with the resolution of the views (number of data points per view) and the matching process should not be sensitive to the variations in the surface sampling of the views. *Robustness to overlap* implies that the correspondence algorithm should not require excessive overlap between adjacent views and should not make any assumptions about their regions of overlap (for instance the DARCES algorithm⁶ assumes a rectangular block of overlapping regions which is not always the case in practice, see Section 3.1). Finally, we define *multi-view correspondence* as a simultaneous one-to-many correspondence approach. More precisely, the correspondence algorithm should be able to simultaneously match a single view with many views as opposed to matching two views at a time. A *pairwise* correspondence algorithm is not practical in case the views are unordered because the algorithm will have to make an exhaustive search for correspondences between $N(N - 1)/2$ pairs of views (where N is the total number of views). The computational complexity of a pairwise algorithm in this case is $O(N^2)$ which makes it computationally expensive.

Few automatic correspondence techniques exist in the literature. However, as will be demonstrated in this paper, all of these techniques are pairwise and achieve automatism by compromising at least one or more of our remaining specifications which limits their applicability. Moreover, there is a lack of a review paper which analyzes these correspondence techniques according to the same set of criteria. On this basis, we present in this paper, a taxonomy of these correspondence techniques and an extensive review of each one of them. We will also perform their analysis according to the above listed important criteria.

The rest of this paper is organized as follows. In Section 2, we give a taxonomy of automatic correspondence techniques. From Section 3 to Section 6, we shall discuss some automatic correspondence algorithms belonging to each category of the taxonomy. These Sections also include the analysis of these algorithms according to our listed criteria. An interesting point to note about these algorithms is that almost all of them rely on the Iterative Closest Point (ICP) algorithm⁴ or one of its variants in order to *verify* the correspondences and *refine* their registration results. The ICP algorithm establishes correspondences between the nearest points of two views and calculates the rigid transformation that would minimize the dis-

tance between these corresponding points. This transformation is applied and the process is repeated until the distance between the corresponding points cannot be reduced any further or no further correspondences can be established between the two views. The ICP algorithm however is not automatic because it requires an initial registration estimate for initialization. In Section 7 we review a complete 3D modeling framework and perform its analysis according to our criteria. Finally, in Section 8 we present our conclusions.

2. Taxonomy of Automatic Correspondence Techniques

Automatic correspondence techniques can be classified as per the taxonomy of Fig. 3 based on the different used approaches³⁰. These techniques can be broadly divided into two categories. The first category is based on an exhaustive search for corresponding points whereas the second category extracts invariant features from the views for the purpose of matching them. Exhaustive search is guaranteed to find the best solution but these techniques are not efficient since the search space becomes very large even for medium size data sets. To roughly estimate the complexity of an exhaustive search algorithm, suppose we want to find correspondences between two views with n data points each. A minimum of three non-collinear corresponding point pairs are necessary to find a rigid transformation (rotation matrix and translation vector) that will align the two views. Therefore, the entire search space to find such pairs would be $C_3^n C_3^n$ which has a complexity of $O(n^6)$.

The invariant feature matching category can be further subdivided into techniques which extract local features and those which extract global features. Global features have two limitations. First, they may not be fully contained inside the region of overlap of the views and second, global features may change from one view to the other due to, for instance, different parts being occluded in the two views. These limitations will result in the difficulty of finding the matching features and hence the correspondences between the two views. Local features, however, have a greater chance of lying fully inside the region of overlap. If a large number of local features are extracted from the views, some of them will be guaranteed to fall inside the region of overlap. Therefore, there are more chances that some local features from one view will match with some of the local features of the other view.

The local feature extraction category is further divided into two sub-categories. The first one extracts local invariant-features by defining a local coordinate basis over the surface of the view while the second one extracts local invariant-features without defining a coordinate basis. In the latter case, it is very challenging to extract invariant features from a view because most of the features that are defined in the coordinates of one view change when transformed to the coordinates of another view. By defining a local coordinate basis over the surface of the view, all of its features (as defined in this coordinate basis) are invariant to rigid transformations. However, the main challenge in this case is to define the local coordinate bases consistently over different views in the presence of noise and variations in surface

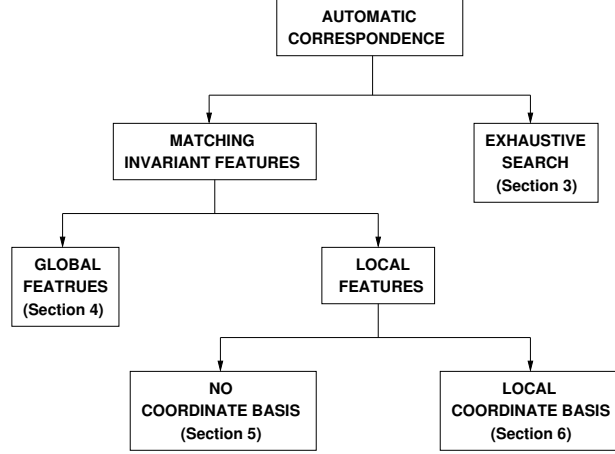


Fig. 3. Taxonomy of automatic correspondence techniques (Figure reproduced from Ref. 30).

sampling. Moreover, a minimum of three *ordered points* are required to uniquely define a 3D coordinate basis. To ensure that a corresponding set of three points is chosen in both the views, 3D bases must be defined from all possible combinations of three points in each view. This leads to a total of P_3^n permutations per view. Therefore, another challenge in this category is to devise a strategy that avoids this combinatorial explosion.

3. Exhaustive Search Techniques

3.1. *RANSAC-based DARCES*

3.1.1. *Description*

RANSAC-based DARCES (Random Sample Consensus based Data Aligned Rigidity Constrained Exhaustive Search)⁶ uniformly samples the scene surface to select three reference points S_p (primary), S_s (secondary) and S_a (auxiliary) such that they form a triangle. Every point in the model is a possible correspondence for S_p . Once a point M_p in the model has been chosen to correspond to S_p , the search for the point M_s as a corresponding point to S_s is constrained to a sphere of radius $d_{ps} = \|S_p - S_s\|$ with its center at M_p (Fig. 4(b)). Similarly, once the point M_s has been chosen the search for a corresponding point to S_a is restricted to a circle with radius d_{qa} and center at point M_q . d_{qa} is the perpendicular distance of point S_a from the line joining S_p and S_s . M_q is the corresponding point to S_q , the perpendicular projection of point S_a on line $S_p S_s$ (Fig. 4(c)).

Once the three corresponding points have been found, a rigid transformation \mathbf{T}_c is calculated and applied to all the reference points. $\mathbf{T}_c = (\mathbf{R}, \mathbf{t})$, where \mathbf{R} is the rotation matrix and \mathbf{t} is the translation vector that align the three scene points (S_a, S_b, S_c) to their corresponding model points (M_a, M_b, M_c) . This transforma-

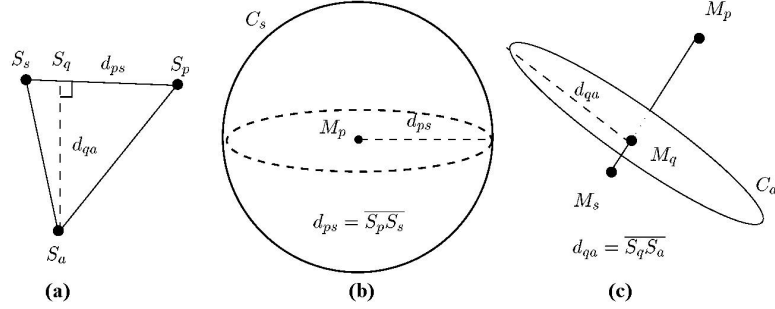


Fig. 4. (a) Triangle formed from the three control points. S_q is the orthogonal projection of S_a to line S_pS_s and d_{qa} is the orthogonal distance from it. (b) The search region for the corresponding point to S_s is a sphere. (c) M_q is the corresponding point to S_q in the model. The search for the corresponding point to S_a is restricted to this circle. (Fig. taken with permission from Ref. 6 © IEEE 1991).

tion is then verified by calculating the distance of the reference points from the model surface. The number of reference points for which this distance is smaller than a threshold is called the overlapping number N_o . N_o is calculated for each possible three point correspondence and the transformation with the largest N_o is considered as the solution. Finally, the ICP algorithm⁴ is applied to refine the registration. To improve the efficiency of the algorithm, more than 3 control points denoted by $S_p, S_s, S_a, S_4, S_5, \dots, S_{n_c}$ are used. The primary control point is selected at random and the remaining control points are selected such that they gradually form a bigger triangle (Fig. 5(a)). Once \mathbf{T}_c has been calculated it is applied to the remaining control points one by one and the transformed point is checked for alignment constraint. If the distance between the transformed scene point and the nearest model point is greater than a preset threshold, \mathbf{T}_c is rejected at an early stage. The algorithm then seeks another set of correspondences for the first three control points. To further improve the efficiency, a coarse to fine search scheme is applied. The correspondence of S_p is first sought at the coarsest level which gives the initial estimate for the next finer level. Once the correspondence of S_p has been hypothesized, the correspondences of the remaining control points are sought at the finest level. Different thresholds of correspondence distances are used for the three levels of search.

To deal with the partially overlapping case, the DARCES algorithm ensures that at least one primary point will be selected such that all the control points lie within the overlapping region. For this purpose, the RANSAC technique is used to calculate the minimum number of iterations required. Suppose the overlapping region (OR) of two partially overlapping views is a square of length l in the index plane of the scene (Figure 5(b)). If all of the control points are to lie inside OR , the primary point must lie in the eroded overlapping region (EOR). The number of expected times of random trials is $E = 1/p$. Where $p = r \cdot (l - d)^2 / l^2$ is the probability of

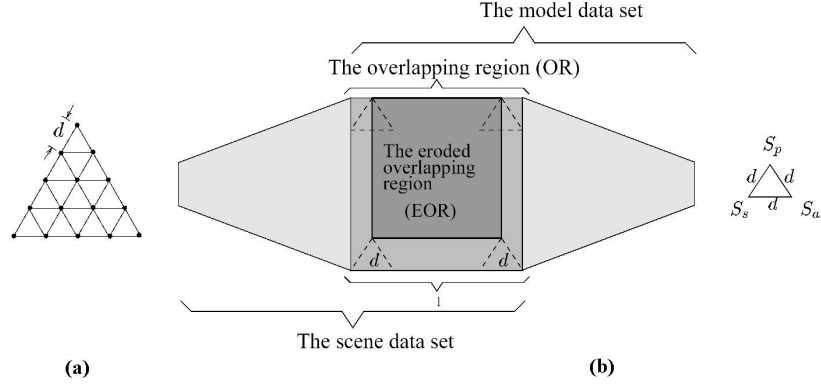


Fig. 5. (a) Fifteen control points selected in the scene index plane. (b) Partially overlapping case. Point S_p must lie in EOR for all the control points to lie inside the OR (Fig. taken with permission from Ref. 6 © IEEE 1991).

a primary point to lie inside the EOR . r is the overlap ratio ($r = n_o/n_s$ i.e. the number of scene points in OR divided by the total scene points).

3.1.2. Analysis

RANSAC-based DARCES is applicable to free-form objects and is guaranteed to find an accurate solution since it is based on an exhaustive search. However, this makes the algorithm inefficient. For the primary scene point S_p , all points on the model are possible correspondences. In case S_p does not lie inside EOR the entire exhaustive search effort for its corresponding point will be futile. This algorithm is also not robust to resolution and surface sampling. The computational complexity of the algorithm grows exponentially with the increase in the resolution and will very quickly get out of control for denser data sets. The algorithm is also sensitive to variations in surface sampling since it matches scene and model points rather than surfaces.

Another limitation of the RANSAC algorithm is that it assumes the overlapping region to be continuous and to form a rectangular region in the image plane (Fig. 5(b)). In practice, the overlapping region is neither continuous nor as simple as claimed. For instance, it may not be continuous and may include holes (areas of no overlap) due to sensor error or self-occlusions. If one of the 15 control points falls into such a region where the surfaces do not overlap, it will not find a corresponding point in the model and the algorithm will reject a valid T_c on this basis. This is not affordable in an exhaustive search. In the worst case, the two surfaces may overlap in a fashion that it may not be possible to have any triangle of 15 points fully inside a region of overlap. To overcome this situation, the size of the triangle should be kept considerably small resulting in transformations that are very sensitive to noise. This will also increase the number of iterations required before a solution is reached.

The ability of the algorithm to handle noise was not discussed in the paper. We believe that noise will adversely affect the performance of the algorithm since this approach linearly matches the distance between points in the two views. Finally, the DARCES algorithm is unable to perform multiview correspondence between unordered views.

3.2. Graph Matching

3.2.1. Description

The graph matching technique⁸ is based on the principle of invariance of distance measures under rigid transformations. By choosing three points P_1, P_2 and P_3 on one surface and calculating their distances $d(P_1, P_2)$, $d(P_1, P_3)$ and $d(P_2, P_3)$, the conservation of distances are used to find similar points on the second surface that would make a similar triangle. In the simplest case of two surfaces having n similar points, the n feature points at view 1 are connected to form a graph G_1 e.g. a spanning tree, a polygon or a path. Next, an optimal matching graph G' will be derived from the second view by the matching algorithm. The corresponding vertices of the two graphs form the corresponding points from which the rigid transformation, that aligns the two surfaces, is calculated.

Let $G_1 = \{P_1, P_2, \dots, P_n\}$ and $d\{P_i, P_{i+1}\}$ be the distance of the i th edge of G_1 . The first step of the algorithm is to find an edge $Q_i Q_j$ in the second view whose length is close to the first edge $P_1 P_2$ of G_1 (Eqn. 2).

$$\|d(Q_i, Q_j) - d(P_1, P_2)\| \leq \epsilon \quad (2)$$

If Eqn. 2 (where ϵ is a tolerance) is satisfied then there are two possibilities i.e. Q_i may correspond to P_1 or P_2 . Both possibilities will be attempted for all matching edges $Q_i Q_j$ to generate a matching graph to G_1 . Consider the first case i.e. Q_i corresponds to P_1 , find recursively the third vertex Q_k such that the edge $Q_i Q_k$ is close to $P_1 P_3$ and $Q_j Q_k$ is close to $P_2 P_3$. This process is repeated till all the n points have been considered. This will result in a candidate graph $G' = \{Q'_1, Q'_2, \dots, Q'_n\}$ of view 2. The candidate graph G'' that has the minimum k th order error (defined by Eqn. 4) from G_1 will be considered as the solution.

$$\|G'' - G_1\|_k \leq \|G'_i - G_1\|_k, \quad \text{for all } i \quad (3)$$

$$\|G' - G_1\|_k = \sum_{i=2}^{i=n} \sum_{j=1}^{\min(k, i-1)} \|d(Q'_i, Q'_{i-j}) - d(P_i, P_{i-j})\| \quad (4)$$

In Eqn. 3 and Eqn. 4, k can vary from 1 to n . Increasing the value of k lowers the chances that a wrong correspondence will be included in the graph. However, it will also increase the comparison time and the chances of rejecting a node. The rejection of a single node means that the entire subtree that follows it will not have to be visited resulting in an overall time improvement. Before a new point Q_l is

accepted as the $(i + 1)$ st node to the candidate graph G' , two conditions must be satisfied. First, the edge difference of the new edge should be less than ϵ . Second, the accumulated k th order error must be less than the threshold $\delta = n(n - 1)\epsilon/2$ (where n is the number of nodes in G). If none of the remaining points satisfy these two conditions, node Q'_i will also be deleted from G' and the search will continue.

In the case of partially overlapping surfaces, there may not be a corresponding point on the second surface for every feature point of the first surface. The algorithm proceeds by considering that each point in view 1 may be an extra point and assigns it to match a null point. The optimum solution is chosen from the candidate graphs which has the maximum number of matched non-null points. In the case of a tie, the decision is based on the least accumulated error. To achieve computational efficiency, the data set is split into s subsets. The matching algorithm is applied to each subset to find its corresponding matching graph. The corresponding data points found for the first subset are removed once the matching graph for the second subset is sought. This process continues till the matching graphs for all the subsets have been found.

3.2.2. *Analysis*

The graph matching algorithm is applicable to free-form objects only if all the data points are considered for building the graph. Extracting feature points will limit its applicability to those objects which have the required features and the results would also depend upon the consistent extraction of feature points from the two views. In case all points are considered for building the graph, the algorithm will become inefficient since it is based on an exhaustive search. The algorithm also lacks robustness to resolution since the search space for a matching graph in the second view exponentially grows with an increase in its data points. This problem may be solved to some extent by splitting the graph into subsets. However, a significant gain in efficiency can only be achieved if the subset graphs are very small. In which case, the accuracy of the algorithm will suffer. This algorithm is also sensitive to variations in surface sampling because it is based on the matching of data points as opposed to the matching of surface patches.

This technique is only suitable for low resolution data and in the case where one surface is a subset of the other, which is generally not the case in the context of 3D modeling. Although it provides a solution to handle partially overlapping surfaces, the suggested solution does not clarify how many consecutive points and how many total points in the graph G_1 can be considered as extra points. Allowing too many consecutive extra points may again adversely affect the efficiency and accuracy of the algorithm. The authors of this algorithm do not provide any evidence regarding the robustness of the algorithm to noise. We believe that noise will further reduce the efficiency of the algorithm by increasing the number of points in G_1 that should be labeled as extra points. Finally, since the graph matching algorithm suffers from inefficiency in the case of pairwise correspondence, it certainly cannot be applied

to solve the multiview correspondence problem.

4. Matching Invariant Global Features

4.1. Bitangent Curve Matching

4.1.1. Description

Bitangent curve matching⁴⁵ is based on the extraction of invariant characteristics of bitangent curve pairs. Correspondence between these curve pairs gives a crude estimate for an initial registration which is then further refined using the ICP algorithm⁴. This technique is robust to the affine distortions introduced during an uncalibrated scan as long as there is one calibrated scan available.

Bitangent points are defined as point pairs on a 3D surface having a common tangent plane. Bitangent curves are those on which a bitangent plane (a plane which is tangent to two points) can roll (Fig. 6(a)). These curves contain a series of bitangent points. Bitangent points and curves are invariant under Euclidean, affine and projective transformations. Bitangent curves are extracted efficiently by finding the self intersection of a surface in dual space as follows. Tangent planes are calculated for every point X on a surface. The equation of the tangent plane is of the form $aX + bY + cZ + d = 0$. Next a dual point D_x is calculated from the tangent plane as follows. The vector $[a \ b \ c]'$ is scaled to constitute a unit vector that corresponds to the surface normal. The value of d after scaling corresponds to the distance of the plane from the origin. The dual point is the vector $D_x = [a \ b \ d]'$ after rescaling. D_x is calculated for every point on the surface and these D_x points make up a new surface known as the dual surface. Since bitangent points have the same tangent plane, they correspond to the self intersections of the dual surface⁴⁵. Fig. 7 shows the bitangent curves on the face of a statue with one pair highlighted in Fig. 7(c). Bitangent curve pairs can have considerably varying lengths.

Invariant signatures are then extracted from the bitangent curve pairs. A single parameterization is used at both points of the pair and for every value of a chosen parameter, invariants are extracted from the coordinates and derivatives of the two points. In the Euclidean case, an invariant is derived using the angle between the tangent vectors to the bitangent curves. Another invariant is derived by expressing the bitangent curves as a function of the arc length of the longest of the two curves (see Ref. 45 for details).

To find correspondence between two surfaces, the invariant signatures of their longest 15 bitangent curves are matched. It is possible for two matching surfaces to contain only parts of their corresponding bitangent curves (in case of occlusion for instance), the signatures are therefore divided into segments. A matching signature segment corresponds to a pair of segments on the bitangent curve pair of each surface. The endpoints of these curves give four points on one surface and their corresponding four points on the other surface. This information is used to calculate the Euclidean or affine transformation between the surfaces. The transformation is

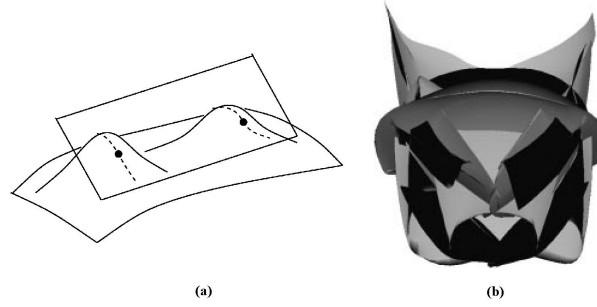


Fig. 6. (a) A rolling plane describing a pair of bitangent curves on a surface. (b) A view of a dual surface (Fig. taken with permission from Ref. 45 © IEEE 1999).

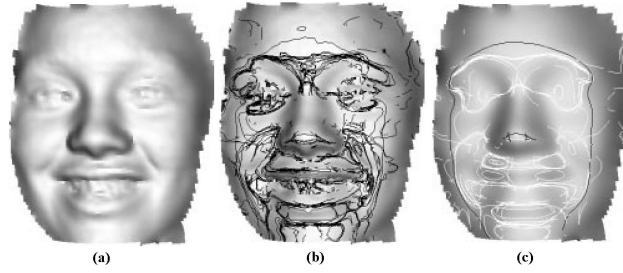


Fig. 7. (a) A 3D view of a face. (b) Bitangent curves on the face. (c) One pair of bitangent curves shown in black (Fig. taken with permission from Ref. 45 © IEEE 1999).

then applied and verified by calculating the closest point distance of the two surfaces for a subset of points. If this distance is less than twice the resolution of the surfaces for a large number of points, the transformation is accepted and finally refined with the ICP algorithm⁴.

4.1.2. *Analysis*

Bitangent curve matching is applicable to free-form objects. The identification of bitangent curves involves the calculation of tangent planes for every point on the scene and model surface. The computational complexity of this process increases linearly with the increase in the resolution of the surfaces. Once the bitangent curves have been identified, the invariant signature extraction and matching would depend upon the number of bitangent curves. A major limitation of this algorithm is that bitangent curves are global features. Global features may not be fully contained inside the region of overlap. In which case, one of the bitangent curves may not be fully inside the overlapping region. This would result in erroneous correspondences. To overcome this problem a lot of overlap must be kept between adjacent views. Another limitation of this algorithm is that the extraction of bitangent curves and

their invariants involve the calculation of derivatives which are sensitive to noise and surface sampling. Therefore, heavy smoothing of the surface is required before calculating the bitangent curves. Smoothing of the surface can itself result in the loss of features (see Fig. 7(c)) leading to variations in the bitangent curves.

5. Matching Local Features Extracted without a Coordinate Basis

5.1. Spherical Attribute Image (SAI) Matching

5.1.1. Description

Spherical Attribute Images were initially developed as a representation scheme for object recognition^{13,19}. However, this approach has also been applied to find correspondences between two partially overlapping surfaces²⁰ by matching their SAI representations. SAI matching falls into the invariant feature matching without local coordinate basis category. This algorithm has two main phases, SAI formation and SAI matching. During SAI formation, the surface data points are approximated by tessellating an arbitrary surface into a semi-regular mesh. The tessellation is deformed repeatedly under three constraints. First, it should be as close to the original surface as possible. Second, all vertices must have exactly three connected neighbours. Third, the mesh should follow the regularity constraint i.e. if P is a node of the tessellation with P_1, P_2 and P_3 as neighbours, G is the centroid of the three neighboring points and Q is the projection of point P on the plane defined by P_1, P_2 and P_3 then Q and G must coincide (Fig. 8(a)). This regularity constraint ensures that the distribution of the mesh nodes on the surface is invariant to rotation, translation and scaling.

Next, the simplex angle is computed at each node. The simplex angle is a measure of curvature and is invariant to rotation, translation and scaling. The simplex angle at P , denoted by $g(P)$, is computed as follows. The node P along with its neighbours P_1, P_2 and P_3 form a tetrahedron inside a sphere of radius r centered at O (Fig. 8(b)). Z is a line passing through O and the center of the circle circumscribed to P_1, P_2 and P_3 ¹³. The cross-section of the tetrahedron with the plane formed by point P and the line Z is a triangle (Fig. 8(c)). The simplex angle (ψ) is the angle between the two edges of the triangle intersecting at P . ψ varies between $-\pi$ and $+\pi$. It is zero for a planar surface, positive for a convex surface and negative for concave surfaces.

These simplex angles are calculated and mapped onto a unit sphere. A unique index is associated with each node which defines a mapping h between the mesh M and a reference mesh S on the surface of the unit sphere i.e. if P is a node on S then $h(P)$ is the corresponding node on M . The mapping h is such that it preserves connectivity. At each node of the unit sphere S , the simplex angle of the corresponding node on the mesh is stored. This results in a spherical image known as the SAI which is invariant to translation and scaling. Moreover, the SAIs of the same surface are identical up to a rotation i.e. one is a rotated version of the other.

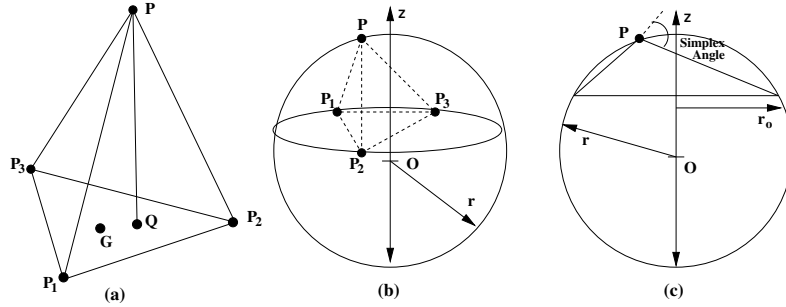


Fig. 8. (a) For local regularity point Q should coincide with point G . (b) and (c) Definition of simplex angle (Fig. reproduced from Ref. 13 © IEEE 1993).

Parts of the original surface with missing data are interpolated by the mesh fitting algorithm and the resulting nodes are marked as interpolated. Nodes are marked as interpolated if their distance from the closest data point is more than a threshold. A complete closed mesh is built even for partially visible surfaces. In such cases the nodes are marked as visible or invisible depending upon their distance from the closest data point. Interpolated and invisible points are not used for matching purposes.

The SAI matching proceeds as follows. Let S and S' be the SAIs of two views of a surface and \mathbf{R} be the rotation that aligns the two SAIs. \mathbf{R} is calculated by minimizing the following cost function:

$$D(S, S', \mathbf{R}) = \sum (g(P) - g(\mathbf{R}P'))^2 \quad (5)$$

The above cost function is calculated for a valid list of correspondences (P_i, P'_j) between the nodes P_i of S and the nodes P'_j of S' . Node P_1 of S is put in correspondence with node P'_{j1} of S' and its two neighbours P_2 and P_3 , are put in correspondence with the neighbours of P'_{j1} i.e. P_{j2} and P'_{j3} respectively. This set of three correspondences define a unique rotation of the spherical image. All possible rotations of the SAI are tried up to its resolution and the one giving a minimum value of D is accepted as the optimum rotation. This rotation, however is not the same as the rotation of the original surfaces and is used to identify corresponding pairs of nodes in the two SAIs. These nodes when mapped back to the original surfaces give the corresponding pairs of nodes on the original mesh. Using these correspondences a transformation is calculated to align both meshes. This transformation is further refined with the ICP algorithm⁴.

To compare the SAIs of different views of a surface, the number of nodes need to be adjusted as the relative size of the visible area depends upon the viewing angle. Let V_1 and V_2 be the two views to be merged and S_1 and S_2 be the number of nodes visible from each viewing angle. Let A_1 and A_2 be their respective visible areas then $S_2/S_1 = A_2/A_1$. This relationship is used to modify the SAIs of different views so that the distribution of nodes in the visible area is consistent between views.

To satisfy this equation some visible nodes have to be moved. The connectivity conservation property of the SAI is used to establish correspondences between the two views of the surface.

5.1.2. Analysis

The applicability of SAI matching is limited to only those surfaces which are free of topological holes²⁰. The original surface is approximated by a tessellating surface. This approximation is not very accurate due to the regularity constraint and may consequently lead to inaccurate correspondences and registration. The SAI preserves relative simplex angle and connectivity information between the nodes, however it does not preserve the relative distance between them. This representation could have been made unique, had the distance between the nodes been recorded. An exhaustive search is carried out for the optimal rotation that would align the two SAIs. This makes the matching process inefficient. The deformation of the tessellating surface and the calculation of the simplex angle is a preprocessing overhead that also affects the efficiency of the algorithm especially in the case of complex surfaces. SAI matching is robust to resolution and surface sampling since it does not match the data points but instead matches the nodes of the tessellating mesh whose resolution and sampling can be controlled.

SAI was originally designed as a representation scheme for object recognition. Therefore, it is not well suited to the partially overlapping case. For example, it always approximates a surface with a closed mesh whether or not it is a closed surface. It then marks each node as visible, invisible or interpolated. It also tries to adjust the number of visible nodes present in the SAIs of different views by marking certain visible nodes in one of the SAIs as invisible. Since there is no prior knowledge of the region of overlap, it is difficult to decide which visible nodes should be marked as invisible. All these factors may lead to an ambiguous representation. The algorithm seems to be sufficiently robust to noise however no experimental results to this effect have been given by the authors. Moreover, this algorithm cannot be used for multiview correspondence between unordered views.

5.2. Roth's Technique

5.2.1. Description

Roth³⁶ proposed an alternate automatic correspondence technique which matches feature points extracted from the intensity images corresponding to the range data. This intensity information is recorded during the acquisition phase along with the range information. The 3D points associated with these feature points extracted from the intensity images are used to find correspondences between two sets of range data. The 3D feature points are interpolated to form triangular meshes. All possible pairs of triangles of the two range data sets are then matched to find possible correspondences. Various rigidity constraints are then applied to prune

incorrect correspondences. The correspondence that aligns the largest number of interest points between two range images is considered as the solution.

A separate broadband camera can be used for a better intensity image since the camera used in range scanners is tuned to the laser frequency. However, Roth discovered that the intensity images provided by most scanners were acceptable for the purpose of extracting feature points. These feature points are computed by finding the local maxima of the intensity gradient⁴⁰ followed by a selection of points which are above a certain threshold. The number of interest points found is a function of the threshold value and the texture of the object. Therefore, this technique is restricted to textured objects. Corners make good interest points due to their maximum intensity gradient in both directions. Figure 9(a) shows the interest points in the intensity component of the range image of a duck.

The 3D points associated with the interest points in the intensity image are used to define 3D triangles using a Delaunay Tetrahedrization (Fig. 9(b)). These triangles are invariant to geometric transformations. If the extracted interest points are the same in the overlapping region of the two range images, then the corresponding triangles will also be the same. The interest points will differ somewhat in practice between images due to the presence of noise and the variations in the surface sampling. However, this technique will succeed as long as there is a single matching triangle in the overlapping region of the two range data sets. This is because one pair of corresponding triangles is sufficient to derive a transformation between the two views. If there are m model feature points and n scene feature points then the number of possible correspondences will be $C_3^m C_3^n$. Since it is not feasible to consider all these possible correspondences, a rigidity constraint is used to limit the search space. Only those triangles which have approximately the same edge lengths are considered as possible matches. This is done by first sorting the edge lengths of each triangle in decreasing order and then quantizing them into k bits. k is chosen such that the smallest bit corresponds to the resolution of the range image. Each triangle can now be represented by a $3k$ bit string. Next, the bit strings of all the triangles are sorted in each data set. All the triangles that have the same bit string are put together in a linked list. This data structure is used to find all the triangles with the same bit string in both views. Such pairs of triangles are considered as potential matches.

A pair of matching triangles gives three sets of matching vertices i.e. 3D points. Each 3D point has its associated normal. The angular difference between the normals of two points is invariant to rigid transformations. Therefore, for a pair of triangles to be matching, the angle difference for their corresponding vertex pairs should be the same. If a vertex pair of two matching triangles does not pass this test, the match is treated as invalid.

Further filtering of incorrect triangle matches is done using the following rigidity constraint. For all feature points in both data sets, the 3D distances of all other feature points are computed. These distances are sorted and placed in a vector called inter-feature distance vector. This inter-feature distance vector should be

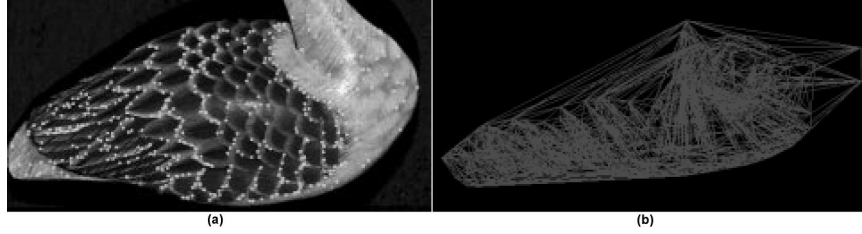


Fig. 9. (a) Feature points in the intensity image of a duck. (b) Delaunay tetrahedrization of the 3D coordinates of the feature points in the range image of the duck (Fig. taken with permission from Ref. 36 © IEEE 1999).

the same for two matching points. The level of compatibility of two feature points will also depend upon their location in the overlapping region of the views. Points that are in the middle of the overlapping region are likely to be more compatible. Two points are considered to match if their inter-feature distance vectors match. Two inter-feature distance vectors are considered to match if they have more than a certain number of similar distances. Only those triangles whose corresponding pairs of vertices pass this test are considered to match. The matching triangle pairs are used, one by one, to calculate a 3D transformation that aligns them. The transformation that aligns the maximum number of interest points of the two range images is considered as the optimal transformation.

5.2.2. Analysis

This algorithm is only limited to those objects which have a significant amount of texture on their surfaces. This is not true for all free-form objects. The success of the algorithm is highly dependent upon the consistent extraction of features from textured objects. This extraction needs to be invariant to the lighting conditions. Moreover, the algorithm also assumes that the range images will be accompanied by their corresponding intensity images from which the feature points are extracted. Not all acquisition systems have this option e.g. imaging radars and laser time of flight scanners. In such cases, a separate camera is required to capture the intensity images of the object. The calibration of this camera with the acquisition system will also require precision.

Apart from these limitations and assuming that the feature extraction is consistent, the algorithm is likely to be accurate. The algorithm is also very efficient since the search space is only limited to the extracted feature points. The algorithm is not affected by the variations in resolution and surface sampling since it only matches feature points and not the entire set of data points in the range images. The robustness of the algorithm to noise depends upon the robustness of the algorithm used to extract the feature points. However, no experimental results were given by the authors to this affect. Finally, this algorithm is for pairwise correspondence and

cannot be used for multiview correspondence between unordered views.

6. Matching Local Features Defined in Local Coordinates

6.1. Matching Oriented Points (*Spin Images*)

6.1.1. Description

In this technique^{26,27} the invariant features are called spin images. Spin images are extracted at each oriented point of the surface. An oriented point is defined as a point along with its normal. A spin image at an oriented point is a 2D histogram of the cylindrical coordinates of its surrounding points (Fig. 10(a)). Spin images are matched using correlation.

The algorithm has two phases. A representation phase in which the spin images are generated for the views and a matching phase during which the spin images of two views are matched to find their correspondences. During the representation phase, the input views are converted into triangular meshes. Normals are then calculated for each vertex point. A spin image, being a 2D histogram of the vertices, is not invariant to different mesh resolutions. Therefore, the mesh is optimized to a uniform resolution. Spin images are then built for each vertex of the model view by making a 2D histogram of the surrounding points within a support size (Fig. 10) and stored in a spin image stack. The technique requires two parameters to be selected, the support size and the bin size of the spin image. Johnson and Hebert²⁷ chose a bin size equal to the mesh resolution and a support size equal to 15×15 bins.

During the matching phase, spin images are generated for a limited number of randomly selected scene points and matched with all the spin images of the model using normalized correlation coefficient. To handle clutter and occlusions the spin images are compared only in bins where both spin images have data. To make the correlation coefficient independent of the number of bins, the variance of the correlation coefficient is included in the calculations. Eqn. 6 gives the similarity measure between two spin images P and Q .

$$C(P, Q) = (\tanh^{-1}(R(P, Q)))^2 - \lambda \left(\frac{1}{N - 3} \right) \quad (6)$$

In Eqn. 6, $R(P, Q)$ is the normalized linear correlation coefficient of spin images P and Q . λ is a weighting parameter which weights the variance against the expected value of the correlation coefficient and is set to 3. N is the number of overlapping bins used. Spin images of corresponding points have a high similarity measure. The similarity measure between all the scene and model spin images are inserted in a histogram (Fig. 11) and the model points which are upper outliers are chosen as the possible corresponding points to each scene point. There are two advantages to this approach. First, if there is no outlier then the scene point has a spin image that is similar to all model spin images and should not be used for correspondence. Second, if more than one outlier exists then this means that there is

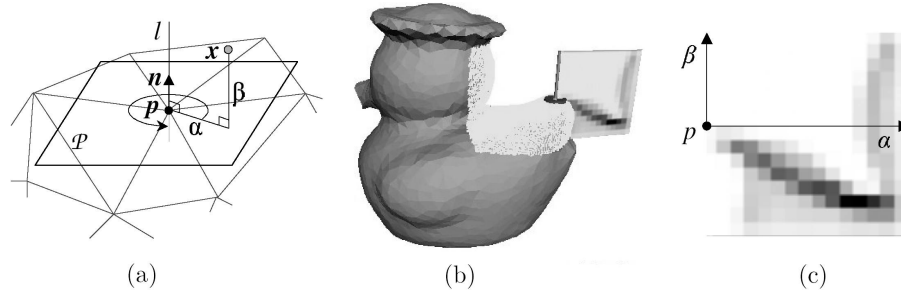


Fig. 10. (a) A triangular mesh with an oriented point p and its normal n . P is the tangent plane at point p . α and β are the cylindrical coordinates of the point x . (b) Generation of a spin image can be assumed as an image plane spinning about the normal of an oriented point and summing points (light shaded region) as they pass through the bins of the image plane. (c) The resultant spin image from (b). (Fig. taken with permission from Ref. 25 © D. Huber 2002).

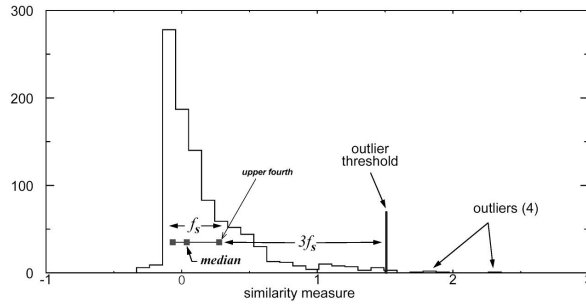


Fig. 11. Histogram of similarity measure between scene and model points (Fig. taken with permission from Ref. 27 © IEEE 1997).

more than one correspondence for the same scene point. All such model-scene correspondences should be considered and the incorrect ones should be filtered out using some other mechanism described below. Outlier correspondences are automatically detected by detecting correspondences that have similarity measures greater than the upper fourth plus three times the fourth spread (f_s) of the histogram (Fig. 11).

The correspondences obtained from the above process contain many incorrect ones which need to be filtered out. The filtration proceeds as follows. Correspondences with similarity measures less than a pre-specified threshold are eliminated. In the next filtering step, two sets of oriented point correspondences (s_1, m_1) and (s_2, m_2) are taken and checked for geometric consistency (Eqn. 7 and Eqn. 8).

$$d_{gc}(C_1, C_2) = \frac{\|S_{m_2}(m_1) - S_{s_2}(s_1)\|}{(\|S_{m_2}(m_1)\| + \|S_{s_2}(s_1)\|)/2} \quad (7)$$

$$D_{gc} = \max(d_{gc}(C_1, C_2), d_{gc}(C_2, C_1)) \quad (8)$$

d_{gc} is the normalized distance between the correspondences $C_1 = (s_1, m_1)$ and $C_2 = (s_2, m_2)$. $S_{m_2}(m_1)$ represents the spin image coordinates (α, β) of the point m_1 with respect to the oriented point m_2 . D_{gc} is the geometric consistency distance. To filter correspondences, $D_{gc}(C_1, C_2)$ is calculated for each correspondence pair C_1 with every other correspondence pair. If $D_{gc}(C_1, C_2)$ is less than a threshold (0.25 in this case) for at least a quarter of the total correspondences, C_1 is accepted otherwise it is removed. The remaining correspondences are grouped into geometrically consistent sets using a criterion W_{gc} (Eqn. 9 and Eqn. 10).

$$w_{gc}(C_1, C_2) = \frac{d_{gc}(C_1, C_2)}{1 - e^{-(\|S_{m_2}(m_1)\| + \|S_{s_2}(s_1)\|)/(2\gamma)}} \quad (9)$$

$$W_{gc}(C_1, C_2) = \max(w_{gc}(C_1, C_2), w_{gc}(C_2, C_1)) \quad (10)$$

W_{gc} is small for geometrically consistent correspondence pairs that are also far apart. Spatially spread correspondences are desirable because they are less sensitive to noise. γ is a normalizing factor and is set to four times the mesh resolution to encourage correspondences that are more than four times the mesh resolution apart. The grouping criteria for a correspondence C is given by Eqn. 11.

$$W_{gc}(C, \{C_1, \dots, C_n\}) = \max_i(W_{gc}(C, C_i)) \quad (11)$$

Given a list of correspondences $L = \{C_1, C_2, \dots, C_n\}$ the grouping proceeds as follows. First, a seed correspondence is selected to initialize a group $G_i = \{C_i\}$. Next a correspondence C_j is searched for in L for which $W_{gc}(C_j, G_i)$ is minimum. If $W_{gc}(C_j, G_i) < T_{gc}$ then add C_j to G_i (T_{gc} is set to 0.25). This procedure is performed for every correspondence in L . It is therefore possible for a correspondence to appear in multiple groups using this algorithm. Once all the correspondences have been distributed among geometrically consistent groups, plausible transformations are calculated for each group. The best transformation is identified by applying the transformation and counting the closest point correspondences in the registered views. The one which gives the maximum number of correspondences is accepted and refined with a variant of ICP⁴⁷.

6.1.2. Analysis

Spin image matching is applicable to free-form objects and is accurate. However, to obtain accuracy it must pass the correspondences given by the spin images through a large number of filtration steps. This is due to the fact that the spin image representation is not unique and gives a large number of ambiguous correspondences. The non-unicity of the spin images occurs due to the mapping of 3D points to 2D images. This algorithm is inefficient since the large number of ambiguous matches have to go through a number of filtration steps. The geometric consistency (filtration) test requires C_2^N pairs of correspondences to be tested (where N is the total number of correspondences that are left after the last filtration step). Similarly the correspondence grouping requires the calculation of W_{gc} between all possible pairs

of correspondences. This process also requires C_2^N operations and may result in an N number of geometrically consistent groups in the worst case. A rigid transformation needs to be calculated for each geometrically consistent group and verified. This verification is computationally expensive because it involves the transformation of the scene view to the coordinates of the model view and the search for neighbouring points between the two views.

Spin image matching is not robust to variations in the resolution and surface sampling of the views. The complexity of the spin image matching increases linearly with the size of the data sets. Moreover, the technique also requires a preprocessing step of converting the meshes into uniform sampling. Apart from these limitations, the algorithm appears to be robust to the amount of overlap between the two views due to two main reasons. First, spin images are local features therefore some of these spin images will lie inside the overlapping region. Second, spin images are matched only in their overlapping bins to cater for occlusions. This algorithm also seems robust to noise because it uses a statistical measure for matching. Statistical measures generally perform better in the presence of noise as compared to linear measures. Finally, this algorithm cannot be used for multiview correspondence between unordered views.

6.2. Three-Tuple Matching

6.2.1. Description

Three-tuple matching⁹ requires the computation of principal curvatures and their direction vectors for the scene and model points. Correspondences are then established between the scene and model points using curvature, distance and direction constraints. The resulting transformations are then verified to identify the optimal one.

The algorithm first selects seed points on the scene surface at which the principal curvatures of the surface can be reliably calculated. Such points are located by fitting a biquadratic polynomial over a $N \times N$ neighbourhood patch around the point and examining the residual fit error. If the error is below a threshold, the point is considered as “bi-smooth”. This process is repeated for all points on the surface and the points are ordered according to the surface fit error. A set of reliable seed points is then selected from this list by choosing the required number of best points such that they are all at least N neighbours away from each other. N is chosen such that the seeds uniformly cover the entire scene surface. The total number of seeds chosen is less than 25.

For each point in the set of reliable seeds say $S_R = \{r_1, r_2, \dots, r_{n_r}\}$, the principal curvatures k_1 and k_2 and their directions \mathbf{t}_1 and \mathbf{t}_2 are calculated (see Ref. 9 for the calculation of curvatures). The vectors \mathbf{n} (normal), \mathbf{t}_1 and \mathbf{t}_2 form an orthogonal frame known as the Darboux frame $\mathbf{\Gamma} = [\mathbf{n} \ \mathbf{t}_1 \ \mathbf{t}_2]$. Next, three seed points s_α , s_β and s_γ are chosen from this set and their corresponding points m_α , m_β and m_γ are sought in the model. Ideally, a set of three corresponding points is sufficient

to calculate the transformation from a scene to the model. s_α is chosen such that it is not umbilical i.e. points having approximately equal principal curvatures. s_β and s_γ are selected such that the three points form a maximum area triangle. The possible corresponding points of s_α are sought in the model by applying the curvature constraint (Eqn. 12).

$$\|k_n(s_p) - k_n(m_p)\| \leq \varepsilon_k \quad n = 1, 2; \quad p = \alpha, \beta, \gamma \quad (12)$$

For each association of s_α and m_α , a rotation matrix \mathbf{R} can be derived by aligning the coordinate frames of s_α and m_α (Eqn. 13).

$$\mathbf{R} = \mathbf{\Gamma}(m_\alpha)\mathbf{\Gamma}^\top(s_\alpha) \quad (13)$$

\mathbf{R} is the rotation matrix that aligns the scene with the model coordinates. \mathbf{R} is prone to errors due to the presence of noise in the range scanner. However, \mathbf{R} can still be used to estimate the likely positions of m_β and m_γ . This is done by combining the distance (Eqn. 14 to Eqn. 16) and direction (Eqn. 17 to Eqn. 19) constraints to limit the search region for these points. This technique is conceptually explained in Fig. 12.

$$d_1 = \|d(s_\beta, s_\gamma) - d(m_\beta, m_\gamma)\| < \varepsilon_d \quad (14)$$

$$d_2 = \|d(s_\alpha, s_\beta) - d(m_\alpha, m_\beta)\| < \varepsilon_d \quad (15)$$

$$d_3 = \|d(s_\alpha, s_\gamma) - d(m_\alpha, m_\gamma)\| < \varepsilon_d \quad (16)$$

$$\mathbf{n}^\top(s_\alpha)\mathbf{p}_1 = \mathbf{n}^\top(m_\alpha)\mathbf{q}_1 \quad (17)$$

$$\mathbf{t}_1^\top(s_\alpha)\mathbf{p}_1 = \mathbf{t}_1^\top(m_\alpha)\mathbf{q}_1 \quad (18)$$

$$\mathbf{t}_2^\top(s_\alpha)\mathbf{p}_1 = \mathbf{t}_2^\top(m_\alpha)\mathbf{q}_1 \quad (19)$$

In Eqn. 17, Eqn. 18 and Eqn. 19, \mathbf{p}_1 and \mathbf{q}_1 represent the vectors from s_α to s_β and from m_α to m_β respectively. To ensure that s_α is selected from the region of overlap, additional constraints are imposed on its selection. Only a subset of seed points (S_R^C) present in S_R will lie inside the overlapping region. However, S_R^C cannot be found exactly due to the lack of prior knowledge of the viewing directions of the scene and model. One approach to this problem is to choose all the points one by one as s_α and find all the possible set of transformations for each association of s_α . However, such an exhaustive search will be inefficient and the set of transformations found may be redundant. Instead of an exhaustive search, a minimal set of seeds S_R^N is identified from the S_R from which s_α can be chosen without resulting in redundant transformations. S_R^N is defined as the set of seeds that would give all the possible transformations found in the set S_R without redundancy. S_R^N is chosen such that at least one of its seeds is visible from any model viewing direction (see Ref. 9 for details).

Once S_R^N is identified, the seed points s_α , s_β and s_γ are selected from this set. The corresponding points on the model are then hypothesized using the curvature, distance and direction constraints. For each hypothesized transformation, a test

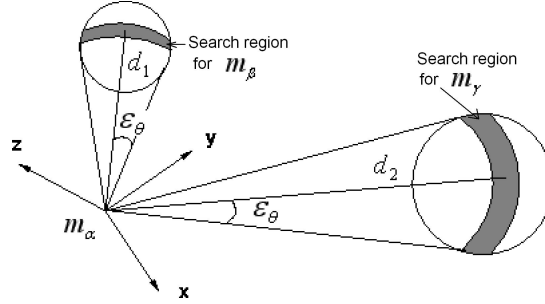


Fig. 12. Point s_α and m_α are aligned. The search region for m_β and m_γ is restricted by ε_θ , d_2 and d_3 . (Fig. reproduced from Ref. 10 © Kluwer Academic Publishers 1997).

is performed to verify whether or not the seed points are visible from the model view point. If the seed points are not visible from the model viewing direction this means that they are outside the overlapping region. Therefore, the hypothesized transformation is rejected. Transformations are calculated for the correspondences of all possible seed points that are visible from the model viewing direction. The transformation that gives the best nearest neighbour correspondences for the overlapping points of the model and the scene is considered as optimal and is accepted. Overlapping and non-overlapping points are distinguished by checking their visibility from both the model and the scene viewing direction. If a visible transformed point does not satisfy the nearest neighbour constraint the transformation is rejected. The optimal transformation is further refined with the ICP⁴ or the Chen and Medioni's algorithm⁷.

6.2.2. Analysis

Three tuple matching is only limited to smooth surfaces since the algorithm calculates the principal curvatures for all points on the model surface. The algorithm is not efficient since the search for reliable seed points in the scene and the calculation of principal curvatures of all the points in the model are both computationally expensive. The algorithm is also sensitive to variations in resolution because its computational complexity linearly grows with the resolution of the views. The algorithm is also adversely affected by any variations in the surface sampling as it is based on matching data points as opposed to surface patches.

This algorithm is not robust to noise since it requires the calculation of principal curvatures which involve the calculation of first and second order derivatives. The error in the calculation of \mathbf{p}_1 for example is likely to be large. Selecting a large value for ε_θ will not reduce the search regions for m_β and m_γ by any significant amount. However, if a small value is selected for ε_θ , the corresponding points of s_β and s_γ may not lie inside the limited search region defined by ε_θ , in which case the accuracy of the algorithm will suffer. The procedure for selecting S_R^N points from

the set of reliable seeds S_R is complicated and unreliable. Once S_R^N is identified the algorithm relies on an exhaustive search and tries all possible combinations of three seed points. The resulting hypotheses of possible transformations is likely to be large and identification of the optimal transformation will again be computationally expensive. Moreover, the 3-tuple matching algorithm is not applicable in the case of multiview correspondence between unordered views.

6.3. Point Signatures

6.3.1. Description

Point signatures¹⁰ is a representation scheme developed for 3D object recognition. It can however be adapted for automatic correspondence. A point signature is a one dimensional signature that describes the surface surrounding a point. It is invariant to rotations and translations and can therefore be used to establish correspondences between two different views of an object.

The signature of a point p is extracted by placing a sphere of radius r centered at p . The intersection of the sphere with the objects surface gives a 3D curve C . The orientation of the curve can be defined by the normal \mathbf{n}_1 , a “reference” vector \mathbf{n}_2 and a cross-product of \mathbf{n}_1 and \mathbf{n}_2 . This forms the coordinate frame of point p ($\mathbf{\Gamma} = [\mathbf{n}_1 \ \mathbf{n}_2 \ \mathbf{n}_3]$). \mathbf{n}_1 is the normal of a plane fitted to the space curve. This plane is translated to the point p in the direction of the normal \mathbf{n}_1 , forming a new plane P' . Next the curve C is projected perpendicularly to the plane P' forming a new 2D curve C' . This projection of points from C to C' forms a signed distance profile, the starting point of which is defined by the vector \mathbf{n}_2 . \mathbf{n}_2 is defined as the unit vector from p to c' (c' is a point on C' that gives the largest positive distance from its corresponding point on C). This distance profile (represented by $d(\theta)$ where $0^\circ \leq \theta \leq 360^\circ$) is known as point signatures and is formed by representing every point on C by the signed distance d from itself to the corresponding point C' and a clockwise rotation of angle θ about \mathbf{n}_1 from the reference direction \mathbf{n}_2 (Fig. 13). C is sampled at intervals of $\Delta\theta$ (set to 15°) making the distance profile a discrete set of values $d(\theta_i)$ for $i = 1, 2, \dots, n_\theta$ (n is the number of samples).

The point signatures of two views are linearly matched to establish point correspondences. In the simplest case, a constant tolerance of ε_{tol} can be used to deal with noisy data. However, in practice there are two additional sources of errors introduced due to the resolution of the sensor and the discrete sampling of the signature. Therefore, ε is chosen dynamically as the sum of all the three possible errors (see Ref. 10 for details). The point signatures $d_s(\theta_i)$ of a scene point is matched with the point signatures $d_m(\theta_i)$ of a model point according to Eqn. 20.

$$\|d_s(\theta_i - \theta_j) - d_m(\theta_i)\| \leq \varepsilon_{tol} \quad \forall i = 1, 2, \dots, n_\theta \quad (20)$$

In Eqn. 20, θ_j is an element of the set of discrete local maxima positions in case $d_s(\theta_i)$ has more than one point which is at maximum (and equal) distance from the curve C . The two points are considered matching as long as there is one value of θ_j

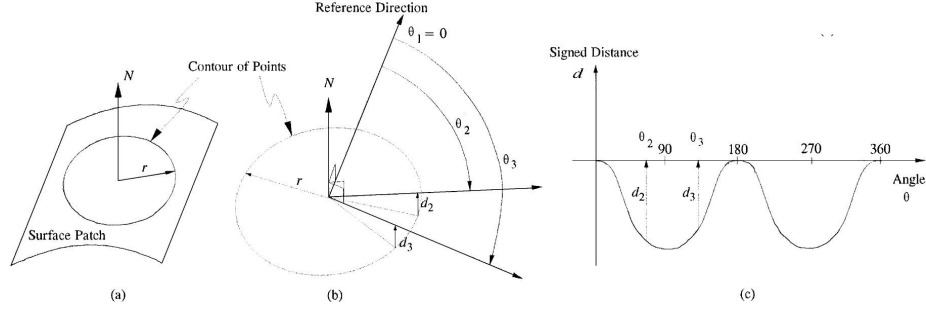


Fig. 13. (a) Contour of points at radius r (b) reference direction (c) point signature (Fig. taken with permission from Ref. 10 © Kluwer Academic Publishers 1997).

for which the above condition is satisfied. A more detailed correlation fit is applied to these points by finding the θ_{opt} which gives the minimum error between the two signatures. If θ_{opt} is non-zero, the reference vector \mathbf{n}_2 of the scene point has to be rotated to \mathbf{n}'_2 about the normal vector \mathbf{n}_1 . The rotation matrix \mathbf{R} , that aligns the scene coordinates to the model coordinates is calculated using Eqn. 21.

$$\mathbf{R} = \mathbf{\Gamma}(m)\mathbf{\Gamma}^\top(s) \quad (21)$$

In Eqn. 21, $\mathbf{\Gamma}(m)$ and $\mathbf{\Gamma}^\top(s)$ are the matrices of coordinate frames of the model and scene point respectively. If the scene point and the model point are correctly associated and the extraction of point signatures is error free then \mathbf{R} will correctly represent the rotation matrix that transforms the scene to the model coordinates. However, due to the presence of noise in the range scanner and the variations in surface sampling, the signatures of corresponding points in two different views will not be exactly similar. Due to this reason, \mathbf{R} is only used as an estimate to limit the search for other correspondence pairs as described in Section 6.2 (Fig. 12).

Due to the presence of symmetries in the object, a single point signature of the scene may match with multiple signatures of the model. Consequently, the resulting rotation matrices are all tried one by one. Points, which do not have a well defined maximum of the distance profile of the curve C will result in an ambiguous coordinate frame. Such points are only considered for matching and not for extracting \mathbf{R} to define search directions for other correspondence pairs. To speed up the search, the model signatures are indexed by their maximum and minimum values so that only those points that have approximately the same maxima and minima are matched. To reduce the chances of incorrect matches, two signatures are generated at two different radii for each point.

The matching algorithm proceeds as follows. First, point signatures are calculated for all the model points. Next, the point signature for a randomly selected point s_{ap} in the scene is calculated and the model points m_{apq} ($q = 1, \dots, h$; where h is the total number of model points whose signature matched with that of s_{ap}), whose signatures match with that of s_{ap} are extracted. For each association of s_{ap}

and $m_{\alpha pq}$, a rotation matrix \mathbf{R} is derived (Eqn. 21) which is used to limit the search region for the remaining correspondences. Two more seeds $s_{\beta p}$ and $s_{\gamma p}$ are selected from the scene within a threshold distance of t_d from $s_{\alpha p}$ such that the three points form a maximum area triangle. The three points are transformed to the model coordinates using \mathbf{T} . The search regions for the corresponding points of $s_{\beta p}$ and $s_{\gamma p}$ in the model are limited by a cone of tolerance ε_θ (see Fig. 12). More than one set of correspondences $(s_{\alpha p}, s_{\beta p}, s_{\gamma p})$ and $(m_{\alpha pq}, m_{\beta pq}, m_{\gamma pq})$ can be established. The optimal transformation is identified by transforming the scene points to the model coordinates and searching for nearest neighbour correspondences. The one that gives the maximum number of correspondences is accepted.

6.3.2. *Analysis*

This algorithm is applicable to free-form objects but lacks accuracy. The main drawback lies in the representation scheme used. The extraction of the second coordinate basis \mathbf{n}_2 has three problems. First, it is not robust to noise. A single outlier point (as a result of noise) can cause a maximum to be created at a wrong location, resulting in a different signature. Second, it is ambiguous in the case of planar and spherical regions. In which case, all the points on the curve C are at an equal distance. Third, matching the signatures with a phase shift corresponding to each possible value of \mathbf{n}_2 will make the matching process inefficient. Since \mathbf{n}_3 is the cross product of \mathbf{n}_1 and \mathbf{n}_2 , it inherits all the drawbacks from \mathbf{n}_2 . Once a single set of corresponding points is found, the search region for the correspondences of the outstanding points is restricted to a cone by the rotation matrix \mathbf{R} . In our opinion, the rotation matrix \mathbf{R} will be quite inaccurate due to the instability of \mathbf{n}_2 and \mathbf{n}_3 . Hence, the correspondences of the remaining two points may not exist in the search cones defined by angle ε_θ .

This algorithm is sensitive to the resolution of the views because point signatures must be computed for all the model points. The computational complexity of the algorithm increases linearly with the resolution of the views. The algorithm is also sensitive to variations in surface sampling as it matches data points of the two views based on their spatial positions. The point signatures algorithm has not been used for multiview correspondence between unordered views.

6.4. *Geometric Histogram Matching*

6.4.1. *Description*

Geometric histogram matching¹ approximates each surface by a triangular mesh. A geometric histogram is built for each facet of the two meshes which describes the relationship of the facet with its surrounding facets within a specified neighbourhood. Next, local correspondences are hypothesized between the scene and model facets by matching their geometric histograms. Finally, a global surface correspondence is sought by identifying consistent local hypotheses using a probabilistic Hough

transform⁴². A more elaborated description of this algorithm is given below.

A surface S represented by a sample of points $P = \{p_1, p_2, \dots, p_N\}$ is approximated by a triangular mesh $S' = \{t_1, t_2, \dots, t_M\}$ where t_i is a triangular facet of the mesh. No assumptions are made about the actual distribution of the facets over the surface as this is unlikely to be repeatable. In Ref. 1, the triangular meshes were constructed using a reconstruction algorithm developed by Hoppe et al.²³. These meshes are then simplified using Garland's mesh simplification algorithm¹⁷. Subsequently, a geometric histogram is constructed for each facet t_i , which described its relationship with the surrounding facets within a predefined neighbourhood. Figure 14(a) shows the relationship between two facets t_i and t_j . The relationship is characterized by two measurements, the relative angle α between the normals of the two facets and the distance d from the plane in which facet t_i lies to all points on facet t_j . These measurements are weighted by the product of the areas of the two facets and accumulated into a 2-D histogram (Fig. 14(b)). The weight of the entry is spread along the distance axis in proportion to the area of t_j at each distance. The entry is blurred with a Gaussian blurring function to compensate for variations in measurements due to the mesh approximation. The coarseness of the mesh defines the scale of the blurring function. These entries are accumulated for all neighbouring facets to constitute the geometric histogram of facet t_i .

The geometric histogram is invariant to rigid transformations and robust to clutter and occlusions. It depends only on the surface shape and not on the placement of the triangular facets over the surface. This property ensures that the same surface, represented by different triangular meshes, should still have the same geometric histogram. Given two surface meshes S'_A and S'_B , their geometric histograms h_i^A and h_j^B are constructed for all their facets, t_i^A and t_j^B respectively. Pairwise matching is then performed between the histograms of the two surfaces using the following similarity metric.

$$D_{ij} = \sum_{\alpha, d} \sqrt{h_i(\alpha, d)} \sqrt{h_j(\alpha, d)} \quad (22)$$

The facets whose histograms give the best match are considered as corresponding facets. These pairs of corresponding facets are local correspondences and each corresponding pair is used as a hypothesis for the global correspondence between the two surfaces S_A and S_B . A transformation that aligns a pair of corresponding facets can then be calculated. This transformation also provides a constraint on the transformation that aligns the complete surfaces. The verification of the hypothesis for global correspondence is performed using a probabilistic Hough Transform⁴². The verification process proceeds as follows. First, the cumulative error (due to noise and variations in sampling) is determined by statistically modeling the error between the true surface and the facets. This error is then propagated through the estimated transformation and is then integrated into the Hough accumulator.

Since the number of corresponding pairs of facets are very large, only a small proportion of 2-tuples corresponding to the largest paired facets are selected. These

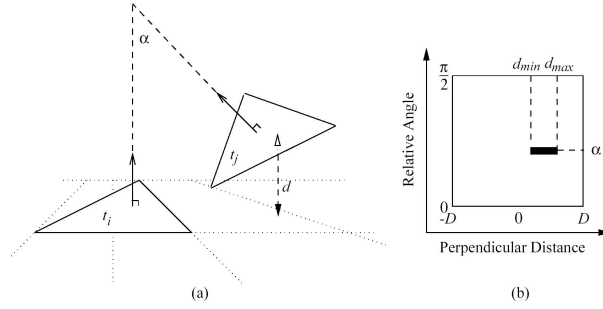


Fig. 14. (a) Geometric relationship between two facets t_i and t_j . (b) The corresponding entry in the geometric histogram of t_i (Fig. taken with permission from Ref. 1 © Springer-Verlag 1998).

2-tuples are used to estimate the rotation component of the transformation and votes are placed in a 3D Hough transform. If a significant peak is found, then 3-tuples are selected using the same criteria, to estimate the translational component of the transformation. The transformation hypothesis which corresponds to a peak in both the rotation and translation spaces is accepted as the optimal transformation for aligning the two views.

6.4.2. Analysis

Geometric histogram matching is applicable to free-form objects. The algorithm lacks details about how the transformation hypothesis verification is exactly carried out. Moreover, Hough transform is equivalent to an exhaustive search and is therefore computationally expensive. Efficiency is gained by considering only a proportion of the 2-tuples with the largest paired facets. However, details were not provided on the exact basis on which this criterion was set. It is possible that the best hypothesis for a transformation could be provided by a 2-tuple that does not lie in the selected proportion. The algorithm does not mention anything about the refinement of the registration results (using for instance an algorithm such as the ICP⁴). From our experience, we have seen that the registration achieved by matching triangular meshes at low resolution (when a mesh simplification algorithm is used) is very crude and needs to be refined. Apart from these drawbacks, geometric histogram matching is independent of the resolution and sampling of the model and scene surfaces. The algorithm also seems to handle noise effectively. The authors however did not provide any proof to this effect. Moreover, this algorithm cannot be used for multiview correspondence between unordered views.

7. An Automatic 3D Modeling Framework

Recently Huber and Hebert^{24,25} proposed a framework for automatic 3D modeling claiming that the framework is capable of automatically generating a 3D model of an object from its unordered views. There are two phases of the framework (as shown

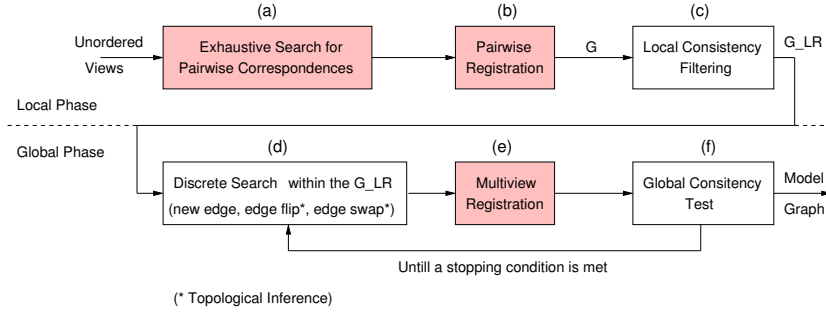


Fig. 15. Block diagram of the automatic 3D modeling framework.

on the block diagram in Fig. 15), the local and the global phase. During the local phase, surface matching is performed on all pairs of views. Since the input views are unordered and a pairwise correspondence algorithm is used for this purpose, the algorithm must exhaustively search for correspondences between $N(N-1)/2$ pairs of views (Fig. 15(a)). Next, these matches are refined using pairwise registration (Fig. 15(b)). The output of these steps is a graph G of relative pose estimates, some of which may be incorrect. Each node in the graph G represents a view and each arc represents the relative pose between its end nodes. A local consistency test (Section 7.3) is then applied to filter out the worst matches from this graph resulting in another graph G_{LR} (Fig. 15(c)). G_{LR} only contains locally consistent matches. However, some of these matches may still be globally inconsistent.

During the global phase, a connected sub-graph in G_{LR} that contains only the correct matches is sought. This phase comprises two different multiview surface matching algorithms (Section 7.4). Each of these algorithms searches for a single edge at a time in G_{LR} , based on various local quality measures (Section 7.1) and adds it to the model hypothesis sub-graph (Fig. 15(d)). Every time a new edge is added to the model hypothesis, it is optimized through the simultaneous multiview registration algorithm (Fig. 15(e)). Next, the model hypothesis is tested for global consistency (Fig. 15(f), Section 7.3). If the newly added edge is globally consistent, it is accepted otherwise it is rejected and another edge in the G_{LR} is sought. This process continues until a stopping condition is met. The output of the global phase is a model graph (which is a spanning tree^a) containing the absolute poses of all the views as a possible solution.

The framework relies on off-the-shelf algorithms for pairwise surface matching, pairwise registration and simultaneous multiview registration (shaded blocks of Fig. 15). A modified version of the spin images technique (Section 6.1) is used for pairwise surface matching. However, any other automatic correspondence algorithm can be used e.g. the ones discussed in this paper. This modified version

^aA tree is a graph in which any two nodes are connected by exactly one path and a minimum spanning tree is a minimum weight tree which includes all the nodes of the graph.

builds spin images from the mesh faces instead of vertex points. The framework uses Neugebauer's algorithm³¹ for pairwise and simultaneous multiview registration. The surface matching algorithm and the registration algorithm are modular components in the framework and can be replaced by better techniques.

In this section, we shall concentrate on the remaining part of the 3D modeling framework (the unshaded blocks of Fig. 15). This framework defines local and global quality measures for surface match verification. Local consistency filtering (Fig. 15(c)) is performed on the basis of local quality measures. The global quality measures are used by the multiview surface matching algorithms for global consistency test in order to filter out globally inconsistent matches (Fig. 15(d) and (f)). We shall first report the local and global quality measures followed by the local and global consistency tests. Finally, we will describe the two variants of multiview surface matching algorithms which use these reported consistency tests to find a solution model graph.

7.1. Local Quality Measures

The local quality measures are used for pairwise surface match verification and filtering. The behavior of the system is modeled for a given sensor and scene type by determining the statistical model of related training data. From these statistical models, the probability of a correct match is estimated given a vector of local quality measures. The local quality measure of a match between surfaces S_i and S_j with relative pose T_{ij} is defined by the generic Eqn. 23.

$$Q_L = \log \left(\frac{P(x|M^+)P(M^+)}{P(x|M^-)P(M^-)} \right) \quad (23)$$

In Eqn. 23, M^+ stands for a correct match and M^- stands for an incorrect match. The probability distributions are estimated from labeled training data and are approximated using Gamma distributions. Three types of local quality measures are used. The first one is based on overlap whereas the remaining two are based on visibility consistency. The local quality measures are calculated as follows. Note that we have used different abbreviations from Ref. 25 for the local quality measures in order to make them more meaningful.

7.1.1. Overlap Quality Measure

This quality measure is based on the overlap fraction F_{OV} and overlap distance D_{OV} (Eqn. 24). F_{OV} is the maximum proportion of each surface that lies in the overlapping region whereas D_{OV} is the root mean square distance between a set of closest point pairs in the overlapping region. These values are computed by sampling K points each from the surface S_i and S_j .

$$Q_{LOV} = \log \left(\frac{P(D_{OV}|M^+)P(F_{OV}|M^+)P(M^+)}{P(D_{OV}|M^-)P(F_{OV}|M^-)P(M^-)} \right) \quad (24)$$

7.1.2. Visibility Consistency-based Quality Measure

The quality measures which are based on visibility consistency are the free space violation (*FSV*) and the occupied space violation (*OSV*). *FSV* occurs when a region of S_j blocks the visibility of S_i from the sensor (Fig. 16(b)). An *OSV* occurs when a region of S_j is not seen by the sensor whereas it should have been (Fig. 16(c)). Both these measures are calculated twice. The first time by transforming surface S_i to the coordinate system of S_j and, vice versa, the second time. These measures are then added up. *FSV* can only occur due to an incorrect match. However, an *OSV* can also occur in cases of correct matches. For example, a sensor can miss a point due to poor reflection or self occlusion.

Two quality measures are derived from the *FSV*. The first one FSV_{raw} uses the raw depth differences in the overlapping regions. If $D = D_{i,j}(1), \dots, D_{i,j}(K)$ constitutes the vector of depth differences between the merged surfaces S_i and S_j , then FSV_{raw} is defined by Eqn. 25.

$$Q_{L_{FSV_{raw}}} = \log \left(\frac{\prod_{k=1}^K P(D_{i,j}(k)|M^+) \prod_{k=1}^K P(D_{j,i}(k)|M^+) P(M^+)}{\prod_{k=1}^K P(D_{i,j}(k)|M^-) \prod_{k=1}^K P(D_{j,i}(k)|M^-) P(M^-)} \right) \quad (25)$$

The second *FSV* quality measure is calculated in a similar manner to Eqn. 24 except that the distances are now calculated along the line of sight of the sensor. Two measures are calculated, range overlap fraction F_{ROV} (Eqn. 26) and *FSV* distance D_{FSV} (Eqn. 27).

$$F_{ROV} = \frac{A_{OV}}{A(S_i)} \quad (26)$$

$$D_{FSV} = \sqrt{\sum_{k \in K_{ij}} w_k (D_{i,j}(k))^2} \quad (27)$$

In Eqn. 26, A_{OV} is the visible overlapping area of the two surfaces and $A(S_i)$ is the visible surface area of S_i . In Eqn. 27, the weight $w_k = A_z(k)/A_{OV}$. The *FSV* local quality measure from the viewpoint of sensor i is calculated using Eqn. 28. A similar quality measure is calculated from the viewpoint of sensor j and the results are added up (Eqn. 29).

$$Q_{L_{FSV}}(S_i, S_j, T_{ij}) = \log \left(\frac{P(D_{FSV}|M^+) P(F_{ROV}|M^+) P(M^+)}{P(D_{FSV}|M^-) P(F_{ROV}|M^-) P(M^-)} \right) \quad (28)$$

$$Q_{L_{FSV}} = Q_{L_{FSV}}(S_i, S_j, T_{ij}) + Q_{L_{FSV}}(S_j, S_i, T_{ji}) \quad (29)$$

The final quality measure is defined by incorporating the *OSV* into Eqn. 28. Any pixel that is defined in one range image and is undefined in the other range image is known as an *OSV* unless it falls outside the sensor's viewing volume. Although even correct matches are expected to have some *OSVs*, incorrect matches generate many more *OSVs* than the correct ones. An *OSV* based feature called *OSV* fraction (F_{OSV}) is calculated using Eqn. 30. The *FSV-OSV* quality measure

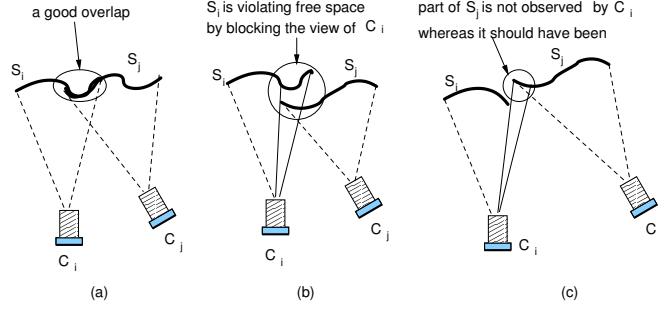


Fig. 16. (a) Both surfaces are close wherever they overlap (b) A case of FSV. S_j is violating free space. (c) A case of OSV. S_j is violating open space. (Fig. reproduced with permission from Ref. 25 © D. Huber 2002).

from the viewpoint of sensor i is given by Eqn. 32. A similar quality measure is calculated from the viewpoint of sensor j and the two are added.

$$F_{OSV} = \frac{A_{OSV}}{A_{OV}} \quad (30)$$

$$A_{OSV} = \sum_{k \in K_{OSV}} A_z(S_j, k) \quad (31)$$

$$Q_{L1_{FSV-OSV}} = \log \left(\frac{P(D_{FSV}|M^+)P(F_{ROV}|M^+)P(F_{OSV}|M^+)P(M^+)}{P(D_{FSV}|M^-)P(F_{ROV}|M^-)P(F_{OSV}|M^-)P(M^-)} \right) \quad (32)$$

In Eqn. 31, K_{OSV} is the set of OSV pixels from range image R_j and $A_z(S_j, k)$ is the z-buffer depth value of pixel k in the z-buffer for S_j .

7.2. Global Quality Measures

The global quality measure is defined as the sum of the local quality measures of all the views that are connected in the graph (Eqn. 33). The views may not be adjacent.

$$Q_{G1}(G) = \sum_{(i,j) \in E_c} Q_L(V_i, V_j, T_{ij}) \quad (33)$$

E_c is the set of connected edges in the graph. Each one of the local quality measures (overlap, FSV_{raw} , FSV and $FSV-OSV$) has a corresponding “sum-local” global quality measure. The statistical quality model used for local quality measure cannot be applied in this case because of varying path lengths. Therefore, a separate statistical model is learned for each path length. Moreover, since the prior probabilities $P(M^+)$ and $P(M^-)$ can no longer be estimated from the data, they are set to 0.5 each.

The effects of topological inference (see Fig. 15(d)) and multiview registration on the statistical model are also taken into account. Topological inference establishes new links in the model graph, thereby shortening the path lengths between non-adjacent views. These new links are used by multiview registration to improve the

alignment of all the views. A second global quality measure is also defined based on a point-to-point multiview registration cost function (Eqn. 34).

$$Q_{G2}(G) = - \sum_{(i,j) \in V_c} \sum_k ((d_{ijk})^2 - \lambda E[(d_{ijk})^2]) \quad (34)$$

In Eqn. 34, d_{ijk} is the distance between the k th correspondence and $E[(d_{ijk})]$ is the expected value of this distance which depends upon the sensor used. It is estimated from multiview registration of a set of training values that are known to be correct. $\lambda \geq 1$ is a weighing parameter.

7.3. Local and Global Consistency Tests

For each quality measure a local consistency test is defined by Eqn. 35 (where λ is a user defined threshold). The local consistency test is used to filter out matches that are of very low quality giving out a model graph G_{LR} . Similarly, the global consistency test $C_G(G)$ for a model graph G is defined by Eqn. 36.

$$\begin{aligned} C_L &= \text{true} && \text{if } Q_L > \log \lambda \\ &= \text{false} && \text{otherwise} \end{aligned} \quad (35)$$

$$\begin{aligned} C_G(G) &= \text{true} && \text{if } \forall_{(i,j) \in E_c} C_L(V_i, V_j, T_{ij}) = \text{true} \\ &= \text{false} && \text{otherwise} \end{aligned} \quad (36)$$

7.4. Multiview Surface Matching

Multiview surface matching is performed during the global phase (Fig. 15(d), (e) and (f)). Two different approaches are used to find the best model hypothesis from the graph G_{LR} , iterative addition algorithm and the iterative swapping algorithm. The iterative addition algorithm starts with the model graph containing only nodes and no edges and constructs a minimum spanning tree using the local quality measures. The spanning tree is built with the Krushkal's algorithm or the Prim's algorithm¹¹. Once initialized with the best edge, both these algorithms connect edges based on the local and global consistency measures. The major difference between the two is that the former considers both adjacent and non-adjacent edges whereas the latter only considers adjacent edges.

The iterative swapping algorithm uses the global quality measure of Eqn. 36 to search for the best model hypothesis in G_{LR} . It uses an additional edge called visibility. Hidden edges indicate division between parts. The algorithm begins by choosing a globally consistent model hypothesis which is a spanning tree. Any spanning tree can be chosen by initially labeling all the edges as hidden. However, a better approach was reported that uses the output of the iterative addition algorithm as an input hypothesis for the swapping algorithm.

At each iteration, the swapping algorithm performs either an edge flip or an edge swap. Edge flip chooses an edge and inverts its visibility state. Edge swap removes

an existing edge from the hypothesis and adds another one (with visibility set to true) which re-establishes the spanning tree. Topological inference is used to find all overlaps and simultaneous multiview registration is applied to this augmented graph. The graph is tested for global consistency as explained in Section 7.3 and if it is globally consistent, its global quality measure is computed. If the new global quality measure is better, the change (edge flip or swap) is accepted otherwise it is rejected.

Two approaches can be applied to edge selection for flip or swap operation. The first approach is greedy edge swapping which tries all edge flips first followed by all edge swaps until no improvement can be found. This approach quickly gets out of control for larger graphs. The second approach is a better one and uses stochastic techniques to choose between flip or swap operations as well as which edges to operate upon. Edges can be selected with uniform randomness or using edge weights based upon edge quality. Moreover, simulated annealing is used in order to avoid local minima. With simulated annealing, an update (edge flip or swap) may be accepted even though it decreases the global quality measure. The probability of accepting such an update is a function of the amount of change in quality and the “temperature” T . Initially a large T is selected which is reduced over time according to a “cooling schedule” (see Ref. 24 for more details).

7.5. Analysis

This framework is applicable to free-form objects. However, it requires an off-the-shelf automatic pairwise correspondence algorithm for initialization. This algorithm must exhaustively find correspondences between $N(N-1)/2$ pairs of views which has a complexity of $O(N^2)$ (N is the total number of views). The computational cost resulting from this exhaustive search, is an extra overhead to the remaining computational cost of the framework. The number of possible edges E in the initial graph is $O(N^2)$ and the complexity of the iterative addition algorithm is $O(EN^3)$. This makes the computational complexity of the iterative addition algorithm $O(N^5)$. For better results, the author suggests that a weighted random swap algorithm be initialized with the output of iterative addition algorithm. This would make the process even more computationally expensive.

Moreover, the multiview surface matching algorithms (iterative addition and iterative swapping) of this framework are not equivalent to a multiview correspondence algorithm according to our definition (Section 1). These algorithms search for discrete edges (pairwise correspondences) in the graph G_{LR} and then verify them after global registration. This contradicts our definition of multiview correspondence i.e. a single view should be simultaneously matched with multiple views.

Another limitation of the framework is that it requires a learning stage in which the behavior of the system is modeled with training data. The training data must be in accordance with the actual scene being modeled e.g. modeling terrains will require training data of terrains and small indoor objects will require training data of small objects. Moreover, the same sensor used for the training data must be

used for the actual scene. These factors limit the applicability of the system as the learning process has to be repeated each time the type of scene or the sensor used is changed. Sometimes it is required to register views acquired with different sensors. In such cases, it will not be possible to train the system if all the sensors are not available during the modeling phase.

Apart from these limitations, the framework is independent of the resolution of the views and surface sampling as long as it uses a surface matching algorithm instead of a point matching algorithm. The framework is also robust to noise since it uses a probabilistic approach to derive its quality measures. The accuracy of the algorithm depends upon its modular components. Any error in these modules, particularly the initialization of the graph and the registration, will propagate to the final stage of the modeling.

8. Conclusion

In this paper, we devised a taxonomy of automatic correspondence techniques. An extensive review of some of the most important automatic correspondence techniques belonging to each category in our taxonomy was also presented. These techniques were analyzed according to a set of important and practical criteria. Table 1 summarizes the results of our analysis. Our analysis shows that none of the available automatic correspondence techniques fully meets these important criteria (see Table 1), which limits their practical applicability. Note that none of these algorithms can be applied to solve the multiview correspondence problem between unordered views. Although the framework discussed in Section 7 is claimed to have the capability of automatically generating 3D models from unordered views, it makes an exhaustive search for correspondences between the unordered views ending up to be computationally expensive. This review concludes that there is still room for work in this area and the need for an automatic correspondence technique that meets all our specifications in order to make the process of 3D modeling efficient and fully automatic.

As it is not possible to cover all automatic correspondence techniques in our survey, a list of additional techniques is provided for completeness. Krsek et al.²⁸ used differential invariants for the automatic identification of correspondences between overlapping views. Godin et al.¹⁸ demonstrated the automatic registration of rotationally symmetric objects based on the distribution of colour patterns. Similar work was performed by Wyngaerd and Gool⁴⁴ who used invariant texture regions for automatic registration of symmetric objects. Silva et al.³⁹ used a novel Surface Interpenetration Measure (SIM) for automatic registration of range images with small overlaps. It is also worthwhile to mention that pairwise registration refinement techniques such as Ref. 4,7 and simultaneous multiview registration refinement techniques such as Ref. 32,33,38 assume that the views are coarsely registered. These techniques are not fully automatic and therefore outside the scope of our survey.

Table 1. Analysis of Automatic Correspondence Techniques.

Algorithm \ Criteria	Applicability to Free-form Objects	Accuracy	Efficiency in terms of Time	Robustness to Resolution	Robustness to Sampling	Robustness to Required Overlap	Robustness to Noise	Multiview Correspondence
RANSAC-based DARCES	Y	Y	N	N	N	N	N	N
Graph Matching	Y	Y	N	N	N	Y	N	N
Bitangent Curve Matching	Y	N	N	Y	Y	N	N	N
SAI Matching	N	Y	N	Y	Y	N	Y	N
Roth's Technique	N	Y	Y	Y	N	Y	Y	N
Spin Image Matching	Y	Y	N	N	N	Y	Y	N
3-Tuple Matching	N	Y	N	N	N	Y	N	N
Point Signature Matching	Y	Y	N	N	N	Y	N	N
Geometric Histogram Matching	Y	N	N	Y	Y	Y	Y	N
Huber's Framework	Y	Y	N	Y	Y	Y	Y	N

9. Acknowledgments

We would like to thank the University of Stuttgart¹⁴ for providing the range data of the isis. We would also like to thank the authors of the cited papers, the IEEE, Springer-Verlag and the Kluwer Academic Publishers for allowing us to reproduce their figures. This research is sponsored by the ARC grant number DP0344338.

References

1. A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi. Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms. *European Conference on Computer Vision*, 2:674–686, 1998.
2. R. Benjemma and F. Schmitt. Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique. In *International Conference on Recent Advances in 3D Digital Imaging*, pages 113–120, 1997.
3. P. Besl. *Machine Vision for Three-dimensional Scenes*. Academic Press, 1990.
4. P. J. Besl and N. D. McKay. 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, February 1992.

5. N. A. Borghese and S. Ferrari. A Portable Modular System for Automatic Acquisition of 3D Objects. *IEEE Transactions on Instrumentation and Measurement*, 49(5):1128–1136, 2000.
6. C. Chen, Y. Hung, and J. Cheng. 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, November 1991.
7. Y. Chen and G. Medioni. Object Modeling by Registration of Multiple Range Images. In *IEEE International Conference on Robotics and Automation*, pages 2724–2729, April 1991.
8. J. Cheng and H. Don. A Graph Matching Approach to 3-D Point Correspondences. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(3):399–412, 1991.
9. C. S. Chua and R. Jarvis. 3D Free-Form Surface Registration and Object Recognition. *International Journal of Computer Vision*, 17:77–99, 1996.
10. C. S. Chua and R. Jarvis. Point Signatures: A New Representation for 3D Object Recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.
11. T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
12. B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In *Computer Graphics, SIGGRAPH*, 1996.
13. H. Delingette, M. Hebert, and K. Ikeuchi. A Spherical Representation for the Recognition of Curved Objects. In *IEEE International Conference on Computer Vision*, pages 103–112, 1993.
14. Department Image Understanding. Stuttgart Range Image Database. <http://range.informatik.uni-stuttgart.de/htdocs/html/>.
15. H. Q. Dinh, G. Turk, and G. Slabaugh. A General Surface Approach to the Integration of a Set of Range Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10):1358–1371, 2002.
16. C. Dorai, G. Wang, A. K. Jain, and C. Mercer. Registration and Integration of Multiple Object Views for 3D Model Construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):83–89, 1998.
17. M. Garland and P. S. Heckbert. Surface Simplification using Quadric Error Metrics. In *SIGGRAPH*, pages 209–216, 1997.
18. G. Godin, D. Laurendeau, and R. Bergevin. A Method for the Registration of Attributed Range Images. In *International Conference on 3D Digital Imaging and Modeling*, pages 179–186, 2001.
19. M. Hebert, K. Ikeuchi, and H. Delingette. A Spherical Representation for Recognition of Free-Form Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):681–690, July 1995.
20. K. Higuchi, M. Hebert, and K. Ikeuchi. Building 3-D Models from Unregistered Range Images. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2248–2253, May 1994.
21. A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt. Marching Triangles: Range Image Fusion for Complex Object Modeling. In *IEEE International Conference on Image Processing*, pages 381–384, 1996.
22. H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald J. Schweitzer, and W. Stuetzl. Piecewise Smooth Surface Reconstruction. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 295–302, 1994.
23. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface Reconstruction from Unorganized Points. *Computer Graphics*, 26(2):71–78, 1992.
24. D. Huber and M. Hebert. 3D Modeling Using a Statistical Sensor Model and Stochas-

38 Ajmal Mian et al.

- tic Search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 858–865, June 2003.
25. D. F. Huber. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, The Robotics Institute, Carnegie Mellon University, USA, July 2002. available at http://www.ri.cmu.edu/pub_files/pub3/huber_daniel_2002_1/huber_daniel_2002_1.pdf.
26. A. E. Johnson. *Spin Images: A Representation for 3-D Surface Matching*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, August 1997.
27. A. E. Johnson and M. Hebert. Surface Registration by Matching Oriented Points. In *International Conference on Recent Advances in 3-D Imaging and Modelling*, pages 121–128, 1997.
28. P. Krsek, T. Pajdla, and V. Hlavac. Differential Invariants as the Base for Triangulated Surface Registration. *Computer Vision and Image Understanding*, 87:27–38, 2002.
29. W. E. Lorensen and H. E. Cline. A High Resolution 3D Surface Construction Algorithm. In *Computer Graphics, ACM SIGGRAPH*, pages 163–169, 1987.
30. A. S. Mian, M. Bennamoun, and R. A. Owens. Automated 3D Model-based Free-form Object Recognition. *Sensor Review*, 24(2):206–215, April 2004.
31. P. Neugebauer. Reconstruction of Real-world Objects via Simultaneous Registration and Robust Combination of Multiple Range Images. *International Journal of Shape Modeling*, 3:71–90, 1997.
32. K. Nishino and K. Ikeuchi. Robust Simultaneous Registration of Multiple Range Images. In *Asian Conference on Computer Vision*, pages 454–461, 2002.
33. T. Oishi, R. Sagawa, A. Nakazawa, R. Kurazume, and K. Ikeuchi. Parallel Alignment of a Large Number of Range Images. In *International Conference on 3D Digital Imaging and Modeling*, pages 195–202, 2003.
34. A. Rangarajan, H. Chui, and J.S. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 3(4):425–440, 1999.
35. C. Rocchini, P. Cignoni, P. Pingi, and R. Scopigo. A Low Cost 3D Scanner Based on Structured Light. In *EUROGRAPHICS*, volume 20, 2001.
36. G. Roth. Registering Two Overlapping Range Images. In *IEEE International Conference on 3-D Digital Imaging and Modeling*, pages 191–200, 1999.
37. S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *3DIM*, pages 145–152, 2001.
38. G. C. Sharp, S. W. Lee, and D. K. Wehe. Multiview Registration of 3D Scenes by Minimizing Error between Coordinate Frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1037–1050, 2004.
39. L. Silva, O. R. P. Bellon, and K. L. Boyer. *Robust Range Image Registration Using Genetic Algorithms and the Surface Interpenetration Measure*. World Scientific, 2004.
40. S. Smith and J. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, pages 45–78, May 1997.
41. M. Soucy and D. Laurendeau. A General Surface Approach to the Integration of a Set of Range Views. *Computer Vision and Image Understanding*, 17(4):344–358, 1995.
42. R. S. Stephens. A probabilistic approach to the hough transform. In *British Machine Vision Conference*, pages 55–59, 1990.
43. J. Williams and M. Bennamoun. Simultaneous Registration of Multiple Corresponding Point Sets. *Computer Vision and Image Understanding*, 81(1):117–142, 2001.
44. J. V. Wyngaerd and L. V. Gool. Combining Texture and Shape for Automatic Crude Patch Registration. In *International Conference on 3D Digital Imaging and Modeling*, pages 179–186, 2003.
45. J. V. Wyngaerd, L. V. Gool, R. Koth, and M. Proesmans. Invariant-based Registration

- of Surface Patches. In *IEEE International Conference on Computer Vision*, volume 1, pages 301–306, 1999.
46. L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 24–36, June 2002.
 47. Z. Zhang. Iterative Point Matching for Registration of Free-form Curves and Surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.