FULLY AUTOMATIC REGISTRATION OF MULTIPLE 3D DATA SETS

*Daniel F. Huber, Martial Hebert**

dhuber@ri.cmu.edu, hebert@ri.cmu.edu
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

This paper presents a method for automatically registering multiple three dimensional (3D) data sets. Previous approaches required manual specification of initial pose estimates or relied on external pose measurement systems. In contrast, our method does not assume any knowledge of initial poses or even which data sets overlap. Our automatic registration algorithm begins by converting the input data into surface meshes, which are pair-wise registered using a surface matching engine. The resulting matches are tested for surface consistency, but some incorrect matches may be locally undetectable. A global optimization process searches a graph constructed from these potentially faulty pair-wise matches for a connected sub-graph containing only correct matches, employing a global consistency measure to detect incorrect, but locally consistent matches. From this sub-graph, the final poses of all views can be computed directly. We apply our algorithm to the problem of 3D digital reconstruction of real world objects and show results for a collection of automatically digitized objects.

1. INTRODUCTION

The advent of relatively low-cost, commercially available laser range sensors has greatly simplified the process of accurately measuring the 3D structure of a static environment, driving the need to automate the processing of 3D data. One problem frequently encountered in 3D data processing is registration, the process of aligning multiple 3D data sets in a common coordinate system. In existing applications, registration is accomplished either by hand or through the use of an external position measurement device such as a global positioning system (GPS). This paper introduces a third alternative: automatic registration, which does not require any external measurements or manual intervention. Formally, we want to solve the following problem:

> Given an unordered set of overlapping 3D views of a static scene *and no additional information*, automatically recover the viewpoints from which the views were originally obtained.

We do not assume any prior knowledge of the original viewpoints such as initial pose estimates or even which views contain overlapping scene regions (*overlaps*). This problem is analogous to assembling a jigsaw puzzle in 3D. The views are the puzzle pieces, and the problem is to correctly put the pieces together without even knowing what the puzzle is supposed to look like.

Three dimensional registration problems can be classified along two axes: the number of input data sets (*views*) and whether initial pose estimates are known. A pose estimate is a rigid body transform (e.g., three rotations and three translations) and can be specified for a single view in world coordinates (*absolute pose*) or with respect to a pair of views (*relative pose*). Along the first classification axis, we use the term *pair-wise registration* when registering two views and *n-view registration* when more than two views are involved. Along the second axis, we use the term *registration refinement* when initial pose estimates are known, and *unconstrained registration* when they are unknown.

With this naming convention, the relationship between our registration algorithm and existing registration methods is clear. For example, the well-known iterated closest point algorithm (ICP) is a pair-wise registration refinement algorithm [1]. Extensions of the ICP algorithm to more than two views are n-view registration refinement algorithms [2][3]. Surface matching, a process often used in 3D recognition systems, is an unconstrained pairwise registration algorithm [4][5]. Our automatic registration algorithm occupies the fourth corner of this taxonomy: unconstrained n-view registration.

Our solution to the unconstrained n-view registration problem consists of two main phases: local registration and global registration. In the local registration phase, the

*This research has been supported by a fellowship from the Eastman Kodak Company.

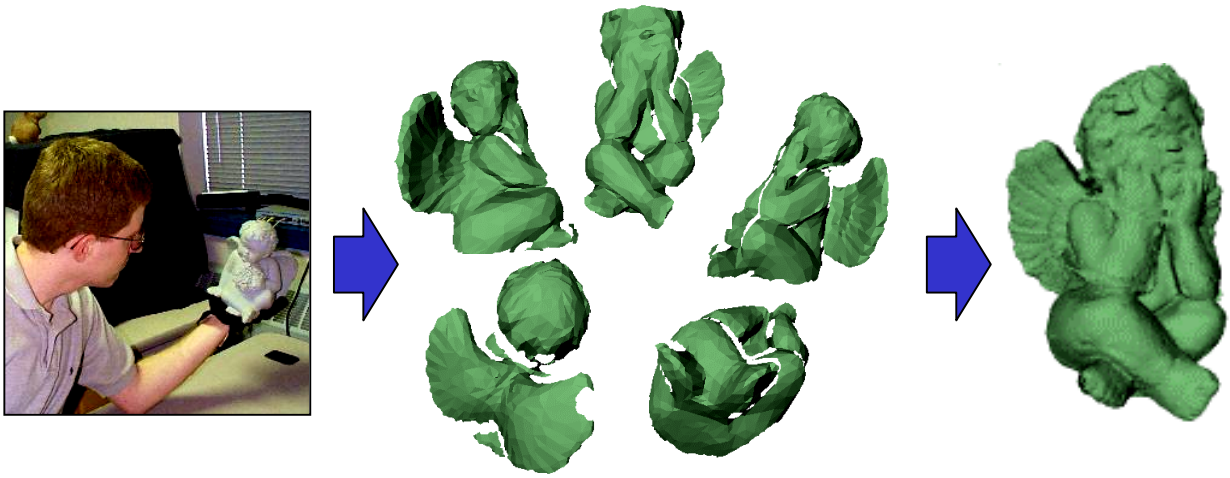

Fig. 1. The 3D digitization application. Holding the object before a laser scanner (left), we obtain 3D data from various viewpoints (center), and automatically construct a digital version of the original object (right).

N input views ($V_i, i \in 1 \ldots N$) are converted to surface meshes ($S_i, i \in 1 \ldots N$), and a surface matching system [6] performs unconstrained pair-wise registration on all view pairs. The resulting matches are verified for surface consistency, but some incorrect matches may be locally undetectable and potential correct matches may be missed. The filtered matches are collected in an undirected graph called the model graph, which encodes the connectivity between overlapping views. In the global registration phase, we search this model graph for a connected sub-graph containing only correct matches. We pose the search as a mixed continuous and discrete optimization problem. The discrete optimization performs a combinatorial search over the space of connected sub-graphs, using a global surface consistency criterion to detect and avoid incorrect, but locally consistent matches, while the continuous optimization adjusts the absolute pose parameters to minimize the distance between all overlapping surfaces, distributing small pair-wise registration errors in a principled way. The final output, the absolute poses of the input views, can be computed directly from the resulting graph[1].

We demonstrate and test our algorithm in the context of 3D object digitization, the purpose of which is to create a 3D digital reproduction of a real-world object (fig. 1). In our application, the object to be digitized is held before a laser scanner while range images are obtained from various viewpoints. We call this hand-held modeling, and it is an exceedingly easy data collection method, requiring no specialized hardware or training and only a few minutes to scan an average object. Alternately, the model can be placed on a table during each scan, or a portable scanner can be moved around while the scene remains stationary. Once data collection is complete, our application produces a digital model of the original object by automatically registering the input views and then merging the registered views into a single entity. Although we illustrate our algorithm with 3D object digitization, the method is general and can be applied in any situation where multiple 3D data sets must be registered.

In the remainder of this paper, we begin by summarizing the related work (section 2). Section 3 provides the necessary background on the model graph concept, and section 4 defines our surface consistency measures. Sections 5 and 6 give the details of the automatic registration algorithm, with section 5 focusing on the local registration phase and section 6 dealing with the global registration phase. Section 7 presents a comparison of three versions of our algorithm on a set of test objects. Finally, in section 8 we discuss the algorithm's limitations and our future work.

2. RELATED WORK

Existing methods for multiple view registration rely on mechanical estimation of poses, manual assistance, or both. One mechanical approach is to mount the scanner on a robot equipped with an absolute positioning sensor. For example, Miller used an autonomous helicopter with a differential global positioning system (DGPS) to construct terrain models [7]. For smaller objects, absolute poses can be obtained by mounting the sensor on a robot arm [8] or by keeping the sensor fixed and moving the object on a calibrated platform [9]. Relative poses can be estimated by mounting the sensor on a robot equipped only with a relative positioning system

[1] In practice, we express the poses with respect to one of the input views, which is selected arbitrarily.

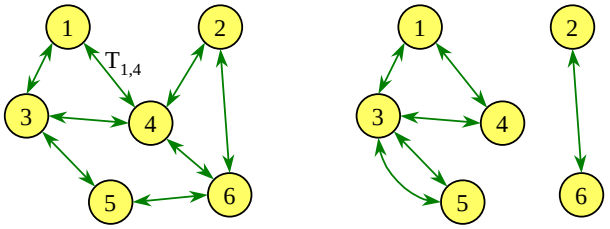

Fig. 2. Example model graphs. A complete model (left) and two partial models (right).

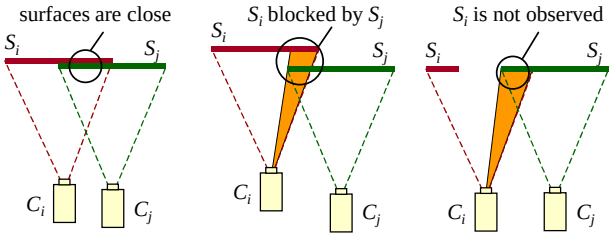

Fig. 3. Visibility consistency from the perspective of C_i: an example of correct registration (left), free space violation (center), and occupied space violation (right).

such as wheel encoders or inertial sensors [10][11][12].

A common manual registration method is to specify corresponding feature points in pairs of range images, from which relative poses can be estimated [2]. In some systems, corresponding feature points are automatically detected and then manually verified for correctness [10]. Alternately, the 3D data can be aligned directly through an interactive method [3]. In more advanced approaches, a person indicates only which views to register, and performs unconstrained pair-wise registration [5][13]. With this approach, the user still must manually verify the registration results.

3. THE MODEL GRAPH

A model graph is an undirected graph G that encodes the topological relationship between views (fig. 2). It contains a node n_i for each input view V_i and an edge $e_{i,j}$ for each pair of overlapping views V_i and V_j. Associated with each edge is a relative pose $T_{i,j}$ and with each node is an absolute pose T_i. The relative pose between two connected views V_i and V_j can be computed by compounding the relative poses along any path from V_i to V_j in G.

A connected model graph specifies a *complete model* and a potential solution to our registration problem, since every view can be transformed into a common coordinate system by compounding relative poses. If, instead, G contains several connected components, each component is called a *partial model*. A spanning tree of G is the minimum specification of a complete model. Additional edges will create cycles in G, which can lead to conflicts because compounding transforms along different paths between two views may give different results. A model is *pose consistent* if the relative pose of two views is independent of the path in G used for the calculation. In practice, pose inconsistencies arise from the accumulation of small errors in relative poses along a path.

4. SURFACE CONSISTENCY

The automatic registration problem would be greatly simplified if we could know with absolute certainty which pairwise matches from our surface matching engine were correct. Such a goal is not attainable, but we can make an estimate by looking at the consistency of the underlying data at the local as well as the global level. Surface consistency is a measure of the degree to which the overlapping data from two (or more) surfaces could represent the same physical surface. A threshold can be used to turn a consistency measure into a classifier. At the local level, we use a continuous valued surface consistency measure to rank pair-wise registration results. At the global level, we use a surface consistency classifier to verify that an entire model (or partial model) is consistent.

4.1. Local surface consistency

For pairs of surfaces, one common measure of surface consistency is the mean squared distance between the overlapping regions (*overlap distance*). However, overlap distance only takes into account the space close to the two surfaces, and in some cases, obviously incorrect matches will have a small overlap distance because the region where they overlap matches well.

For range sensors with a single point of projection, we can develop more powerful measures that take advantage of the sensor's entire viewing volume by looking at the consistency of the two surfaces along the line of sight from each of the sensor viewpoints. We call this concept *visibility consistency*. For example, consider the surfaces in figure 3 viewed from the sensor position C_i. For a correct registration, the two surfaces have similar range values wherever they overlap (fig. 3 left). For an incorrect registration, two types of visibility inconsistencies can arise. A *free space violation* (FSV) occurs when a region of S_j blocks the visibility of S_i from C_i (fig. 3, center), while an *occupied space violation* (OSV) occurs when a region of S_j is not observed by C_i, even though it ought to be (fig. 3, right). Free space violations are so named because the blocking surface violates the assumption that the space is clear along the line of sight from the sensor to the sensed surface. Similarly, OSV surfaces violate the assumption that range sensor detects oc-

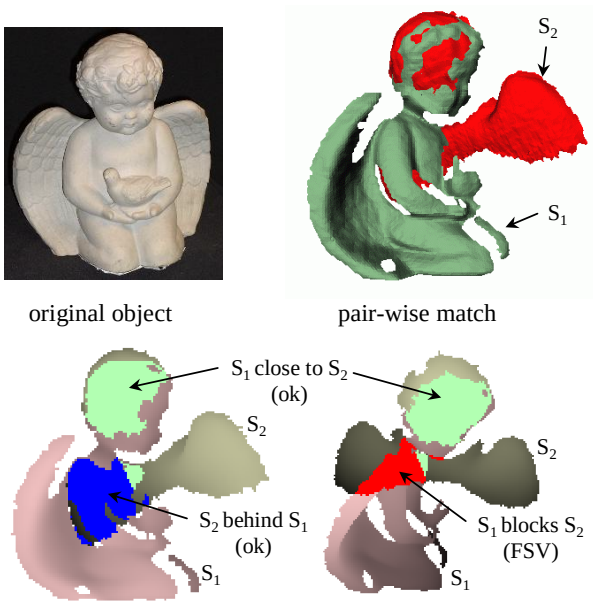

Fig. 4. An example of visibility consistency for an incorrect match. The z-buffers from each viewpoint (bottom row) show the classification of pixels for the FSV fraction consistency measure (eq. 8). The indicated FSV pixels in the z-buffer for view 2 suggest an incorrect match.

cupied space. Here, we focus on FSV's, but the potential of OSV's is discussed in section 8. Visibility consistency has been used previously in other 3D vision contexts, including hypothesis verification [14], surface registration [15], range shadow detection [16], and multi-view integration [17][18].

We can detect FSV's with respect to sensor position C_i by projecting a ray from the center of projection of C_i through a point p on S_i. If the ray passes through S_j at a point q which is significantly closer to C_i than p, then q is an inconsistent point. We must test whether q is *significantly* closer because even for correctly registered surfaces, p and q will not have precisely the same range.

We can efficiently implement FSV detection using two z-buffers [19]. To compute FSV's for surfaces S_i and S_j with respect to C_i, both surfaces are projected into separate z-buffers (Z_i and Z_j) using the coordinate system and parameters of C_i (e.g., focal length, viewing frustum). The depth difference

$$D_{i,j}(k) = Z_j(k) - Z_i(k) \tag{1}$$

is then computed for each pixel $x(k)$ where both z-buffers are defined (fig. 4).

We have developed two local consistency measures based on the FSV concept. The first one, which we call the FSV likelihood, is a statistical measure based on the likelihood ratio test. Given the two possible hypotheses, H^+ (correct match) and H^- (incorrect match), and the set of depth difference measurements $D = \{D_{i,j}(1), \ldots, D_{i,j}(K)\}$, we estimate

$$L(S_i, S_j) = \frac{Pr(H^+|D)}{Pr(H^-|D)} = \frac{Pr(D|H^+)Pr(H^+)}{Pr(D|H^-)Pr(H^-)} \tag{2}$$

Assuming samples of D are independent and taking the logarithm, we have

$$\begin{aligned} \ln(L(S_i, S_j)) &= \sum_{k=1}^{K} \ln Pr(D_{i,j}(k)|H^+) \\ &- \sum_{k=1}^{K} \ln Pr(D_{i,j}(k)|H^-) \\ &+ \ln Pr(H^+) - \ln Pr(H^-) \end{aligned} \tag{3}$$

An independent likelihood ratio $L(S_j, S_i)$ can be computed with respect to sensor viewpoint C_j. Frequently, an incorrect match will be detectable from only one viewpoint, so we conservatively combine $L(S_i, S_j)$ and $L(S_j, S_i)$ to form the FSV likelihood:

$$\overline{L}(S_i, S_j) = -\min(\ln(L(S_i, S_j)), \ln(L(S_j, S_i)) \tag{4}$$

The smaller the value of $\overline{L}(S_i, S_j)$, the more likely it is a correct match[2]. The corresponding FSV likelihood classifier is:

$$\overline{L}_C(S_i, S_j) = \begin{cases} 1 & \text{if } \overline{L}(S_i, S_j) < t_l \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The probabilities in equation 3 can be estimated from labeled training data. We use a set of hand-labeled matches obtained from exhaustive unconstrained pair-wise surface matching of the views of a typical object. First, we compute separate histograms of the depth differences for the set of correct matches (fig. 5, left) and the set of incorrect matches (fig. 5, right). We then model $Pr(D|H^+)$ as a mixture of two Gaussians, one for outliers and one for inliers, fitted to the corresponding histogram. The process is repeated for the incorrect matches to estimate $Pr(D|H^-)$. Mixtures of two Gaussians are necessary because correct matches will contain some outliers, primarily due to small registration errors, and incorrect matches will contain inliers in the region that was matched during surface matching. $Pr(H^+)$ and $Pr(H^-)$ are estimated using the frequency of correct and incorrect matches in the training set.

[2]The sign change is introduced to make this measure consistent with our other measures such as overlap distance.

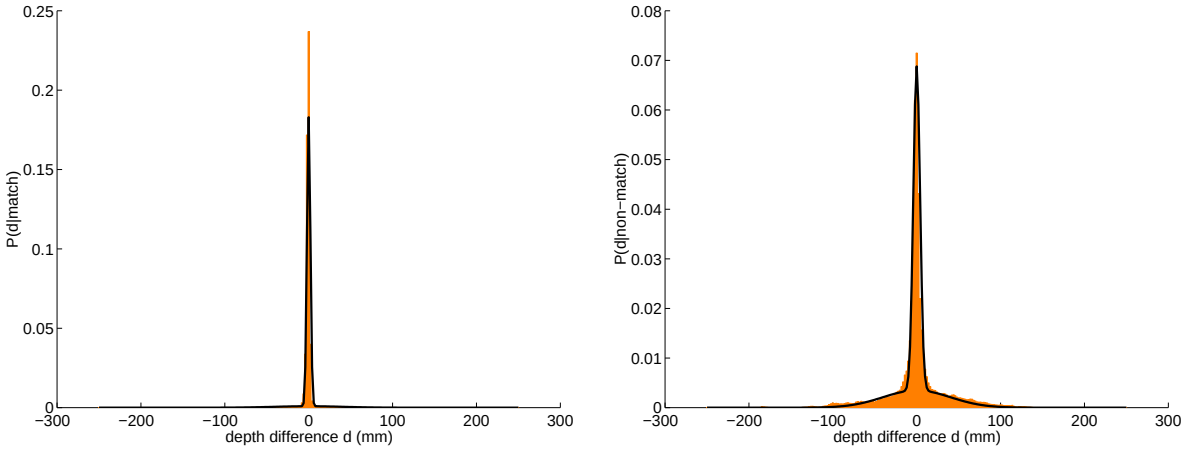

Fig. 5. The distribution of depth difference measurements D over a large set of correct matches (left) and incorrect matches (right) from a test object. The predicted distributions, mixtures of two Gaussians learned from a separate training set, are overlaid (thin black line).

For pair-wise matches, the likelihood ratio test is a good measure of surface consistency, but for non-adjacent views in a model graph, the accumulation of error when computing the relative transforms reduces the accuracy of this test. For this situation, we use an alternative method for estimating surface consistency, the FSV fraction (fig. 4). To compute the FSV fraction, we apply a threshold, t_{SS}, to the depth difference measurements (eq. 1) to classify the overlapping pixels into one of three categories: points on the same surface (X_{SS}), points that are FSV's (X_{FSV}), and "don't care" points where S_j is behind S_i (X_{DC}):

$$x(k) \in \begin{cases} X_{SS}(i,j) & if|D_{i,j}(k)| \leq t_{ss} \\ X_{FSV}(i,j) & if D_{i,j}(k) > t_{ss} \\ X_{DC}(i,j) & if D_{i,j}(k) < -t_{ss} \end{cases} \quad (6)$$

We then compute the fraction of points that are FSV's, ignoring "don't care" points (class X_{DC}):

$$F(S_i, S_j) = \frac{|X_{FSV}(i,j)|}{|X_{FSV}(i,j)| + |X_{SS}(i,j)|} \quad (7)$$

As with the FSV likelihood measure, we can perform the computation with respect to sensor viewpoint S_j to get $F(S_j, S_i)$. Combining the results gives the FSV fraction:

$$\overline{F}(S_i, S_j) = \max(F(S_i, S_j), F(S_j, S_i)) \quad (8)$$

The corresponding FSV fraction classifier is:

$$\overline{F}_C(S_i, S_j) = \begin{cases} 1 & \text{if } \overline{F}(S_i, S_j) < t_f \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

4.2. Global surface consistency

Global surface consistency is the straightforward extension of local surface consistency to an entire model. A model is globally surface consistent if every pair of views is locally surface consistent according to the FSV fraction classifier:

$$C_M(G) = \begin{cases} 1 & \text{if } \forall_{(i,j)\in V_C} \overline{F}_C(S_i, T_{i,j} S_j) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where V_C the set of connected (not necessarily adjacent) view pairs in G, and $T_{i,j}$ is the relative pose computed by compounding transforms along a connecting path between n_i and n_j in G.

5. LOCAL REGISTRATION PHASE

Now that we have defined the model graph and the surface consistency measures, we can fully explain our automatic registration algorithm. The process will be demonstrated on the angel2 test object (fig. 9g).

In the local registration phase, we attempt to register all pairs of views using a surface matching algorithm. For small numbers of views (≈ 20), this exhaustive registration strategy is reasonable. For larger scenes, the combinatorics make this approach infeasible, and view pairs must be selectively registered (see section 8).

In preparation for surface matching, the views are preprocessed as follows: The input range images are converted to triangular surface meshes by projecting into 3D coordinates and connecting adjacent range image pixels. Mesh faces within range shadows (which occur at occluding boundaries in the range image) are removed by thresholding the angle between the viewing direction and the surface normal. For computational efficiency, the meshes are simplified using Garland's quadric algorithm [20].

The surface matching algorithm performs unconstrained pair-wise registration of two surfaces based on their shape.

We treat this process as a black box, which takes two meshes as input and outputs a list of relative pose estimates. Details can be found in [5]. If the two views overlap, the algorithm often finds the correct relative pose, but it may fail for a number of data-dependent reasons (e.g., not enough overlap or insufficient complexity of the surfaces). Even if the views do not contain overlapping scene regions, the algorithm may nevertheless find a plausible, but incorrect, match. Furthermore, symmetries in the data may result in multiple matches between a single pair. The model graph of pair-wise matches for the angel2 data set is shown in figure 7a. For illustration, the matches have been hand-classified, but these labels are, of course, not known by the algorithm.

Next, the alignment of each match is improved by applying a pair-wise registration refinement algorithm. We have implemented two algorithms for this – one based on the ICP algorithm but extended to handle partially overlapping surfaces [21], and a second method that minimizes distances between points and tangent planes in a manner similar to that described by Chen and Medioni [22].

Finally, we perform a local surface consistency test by applying the FSV likelihood classifier to the matches (eq. 10). We classify the matches using a conservative threshold chosen with the intention of eliminating obviously incorrect matches without removing any correct ones. The resulting model graph G_{LR} is shown in figure 7b.

6. GLOBAL REGISTRATION PHASE

The global registration phase uses the locally consistent pair-wise matches (G_{LR}) to construct a pose consistent and globally surface consistent model from which the absolute poses can be read directly. The connected sub-graphs of G_{LR} represent the set of all possible model hypotheses for the given pair-wise matches. To succeed, the global registration must find a sub-graph containing only correct matches; a single incorrect match generally results in a dramatically incorrect solution (fig. 9p).

The global registration can be posed as a mixed discrete and continuous optimization problem over the discrete model sub-graph structure and the continuous valued pose parameters within. We decompose the problem into two nested sub-problems: an inner continuous optimization over absolute poses for a fixed model graph and an outer discrete optimization over model graphs for fixed poses. For the discrete optimization, we sequentially construct a spanning tree from the edges in G_{LR} using a modified version of Kruskal's minimum spanning tree algorithm [23]. Using a spanning tree ensures that the graph is always pose consistent and allows us to directly compute absolute pose estimates. It also reduces the continuous optimization step to an instance of the n-view registration refinement problem. Our n-view registration refinement implementation, based on Neugebauer's [2], minimizes the squared distance between points and their corresponding tangent planes. The correspondences are established between all overlapping view pairs, not just the edges from the current model graph. At the end of each step, the graph is checked for global surface consistency, which ensures the final solution will be surface consistent and reduces the chances that the algorithm will fall into a local minimum. The pseudo-code for this algorithm is shown in figure 6.

```
1: G ← G_0
2: for all edges e_{i,j} ∈ G_LR, sorted in increasing order
   using eq. 4 do
3:   if n_i and n_j are not connected in G then
4:     G' ← G ∪ e_{i,j}
5:     n_view_register(G')
6:     if G' is globally surface consistent (eq. 10)
       then
7:       G ← G'
```

Fig. 6. Pseudo-code for the `full` algorithm for the global registration phase.

Initially, G represents N partial models with one view each (line 1). The edges of G_{LR} are sorted by their FSV likelihood measure and tested one at a time. In each iteration through the loop, the best untested edge from G_{LR} is selected, and if it connects two components, a temporary model graph G' is formed, thereby joining two partial models (line 4). The alignment of the views in G' is improved using an n-view registration refinement algorithm (line 5). If the resulting partial model is globally surface consistent (line 6), the new edge is accepted, and G' becomes the starting point for the next iteration (line 7). Eventually, the algorithm either finds a spanning tree of G_{LR}, resulting in a complete model, or the list of candidate matches is exhausted, resulting in a set of partial models. Figure 7 shows the model graph G at several stages, and the final model, corresponding to the graph in figure 7f, is shown in figure 8.

In addition to the full algorithm described above (`full` hereafter), we tested two simpler versions of our algorithm to analyze the effects of the continuous optimization and the global consistency check. The `discrete_only` algorithm omits the continuous optimization step (line 5), and the `min_span` algorithm skips the global consistency check (line 6) as well.

The `min_span` algorithm, which finds the minimum spanning tree of G_{LR}, has the advantage that it is simple, fast, and always finds a solution, but the result may not be globally surface consistent. The global consistency test can be performed at the end as a verification, but it is not possible to correct the inconsistency.

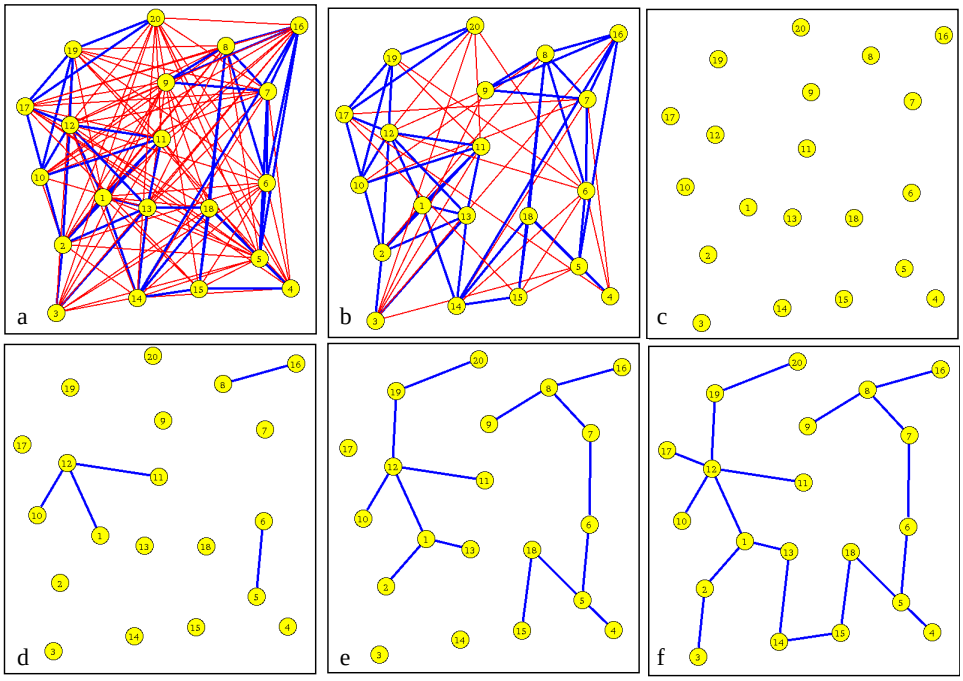

Fig. 7. Model graphs from the local registration phase (a-b) and global registration phase (c-f) for the angel2 test object. Matches were hand-labeled for illustration: thick (blue) edges are correct matches, and thin (red) edges are incorrect matches. a) The model graph from exhaustive pair-wise registration; b) after filtering the worst matches (G_{LR}); c) empty model graph (G_0); d) after 5 steps of `full`; e) after 15 steps; f) the final model graph.

The `discrete_only` algorithm integrates the global surface consistency check into `min_span`, effectively allowing a single step of backtracking. However, the buildup of small pair-wise errors leads to large discontinuities between some overlapping surfaces, which may cause a model to be globally inconsistent even though it contains only correct matches. By incorporating n-view registration refinement at each step, the errors are evenly distributed over the entire model, allowing the `full` algorithm to find the correct solution in some cases where `discrete_only` fails.

7. RESULTS

We tested our automatic registration algorithms by digitizing a collection of ten test objects (fig. 9). Using a Minolta Vivid 700 laser scanner, we obtained 15 to 20 views of each object, scanning with the hand-held data collection method described in the introduction. A black background and glove allow simple, automatic segmentation of the background by thresholding the intensity image. We compared the performance of the three global registration algorithms described in section 6. The results are shown in table 1.

Overall, the results were very good – the `full` algorithm found a qualitatively correct model in nine of the ten test cases. For two test sets (angel2 and letter_y), `min_span` failed where `discrete_only` succeeded. This is because one of the most consistent matches was actually an incor-

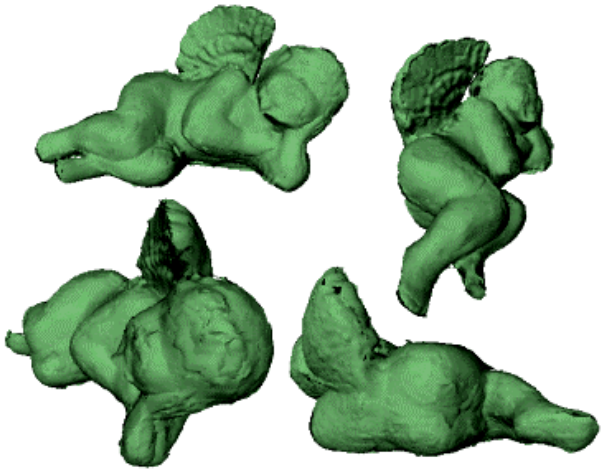

Fig. 8. The automatically registered views of the angel2 test object.

Fig. 9. Photographs of the ten test objects (first and third columns) and the resulting 3D models (second and fourth columns). a-b) angel1; c-d) coder; e-f) dwarf; g-h) angel2; i-j) angel4; k-l) dino; m-n) rabbit; o-p) angel3; q-r) letter_j; s-t) letter_y.

object	min_span	discrete_only	full
angel1	+	x (2 components)	+
angel2	x (1 err.)	+	+
angel3	x (1 err.)	x (1 err.)	x (1 err.)
angel4	+	+	+
coder	+	+	+
rabbit	+	+	+
dwarf	+	+	+
letter_j	+	+	+
letter_y	x (1 err.)	+	+
dino	+	+	+

Table 1. Performance of the three algorithms on the test models. + indicates a correct result (i.e., a complete model with no incorrect matches); x indicates an incorrect result (i.e., partial models or a graph with incorrect matches).

rect match, and the `discrete_only` algorithm correctly detected the incorrect match using the global consistency test. For the angel1 test set, both `min_span` and `discrete_only` failed but `full` succeeded. In this case, the accumulation of pair-wise error prevented `discrete_only` from merging the last two components into a consistent model. The components output by `discrete_only` represented the left and right sides of the object.

In one case (angel3), none of the algorithms succeeded. This is because the pair-wise matching phase did not find any correct matches between two clusters of views: one set representing the front and sides of the object and the other containing views of the back. The global registration process cannot succeed in this situation, but one option is to acquire more data sets that span the boundary region. Another approach is to feed the partial models back into the pair-wise matching phase, treating each partial model as a single view. The greater surface area of each partial model can result in matches that would not have been found in the initial matching phase.

8. FUTURE WORK

We have identified several aspects of our automatic registration method to be further developed. One problem that can arise in the global registration phase occurs when an incorrect match is added to the graph and the resulting partial model is still globally consistent. Once the algorithm proceeds to the next iteration, there is no hope of finding the correct solution. This problem can be addressed in two ways. One is to incorporate backtracking into the sequential algorithm, turning it into a depth first search of the space of all spanning trees. However, it may be necessary to search nearly the entire space of spanning trees if the search chooses an incorrect edge early in the process. A second solution is to turn to a stochastic algorithm.

We have investigated using a RANSAC algorithm, in which spanning trees are randomly sampled from G_{LR} and then evaluated using the global consistency test. Unfortunately, depending on the number and arrangement of incorrect matches in G_{LR}, a very large number of trials may be required. Our next step is to experiment with other stochastic methods such as simulated annealing. The `min_span` algorithm could be used to generate a starting solution for such methods.

Figure 9p shows an example of a model that contains a single incorrect match. Although obviously wrong, the model is actually consistent according to our global consistency test. This situation could be avoided with an enhanced test that considered occupied space violations (OSV's) as well as FSV's. Detecting OSV's requires a more sophisticated sensor model than FSV's because surfaces may go undetected for a number of reasons (e.g., the surface is out of sensor range or the normal is too oblique to viewing direction).

Finally, we must address the issue of view selection. To scale automatic registration to a large number of views, we need to be selective about which view pairs we attempt to register. One approach is to use information inherent in each view to sort the views based on the likelihood of a successful match or to partition into groups that are likely to match with each other.

9. CONCLUSION

We have presented a method for automatically registering a set of 3D views of a scene. The procedure uses a combination of discrete and continuous optimization methods to construct a globally consistent model from a set of pair-wise registration results. We compared three versions of our algorithm, and demonstrated its utility by automatically constructing 3D models of a number of objects.

10. REFERENCES

[1] Paul Besl and Neil McKay, "A method of registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[2] Peter Neugebauer, "Geometrical cloning of 3D objects via simultaneous registration of multiple range images," in *Proceedings of the 1997 International Conference on Shape Modeling and Applications*, Mar. 1997, pp. 130–9.

[3] Kari Pulli, "Multiview registration for large data sets," in *Proceedings of the Second International*

Conference on 3-D Digital Imaging and Modeling (3DIM'99), Oct. 1999, pp. 160–8.

[4] Chu-Song Chen, Yi-Ping Hung, and Jen-Bo Cheng, "RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1229–34, Nov. 1999.

[5] Andrew Johnson, *Spin-Images: A Representation for 3-D Surface Matching*, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Aug. 1997.

[6] Andrew Johnson and Martial Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–49, May 1999.

[7] James Ryan Miller, Omead Amidi, and Mark Delouis, "Arctic test flights of the CMU autonomous helicopter," in *Proceedings of the Association for Unmanned Vehicle Systems, 26th Annual Symposium*, July 1999.

[8] Mark Wheeler, *Automatic modeling and localization for object recognition*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, Oct. 1996.

[9] Greg Turk and Marc Levoy, "Zippered polygon meshes from range images," in *Proceedings of SIGGRAPH 94*, July 1994, pp. 311–18.

[10] S. F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin, "A system for indoor 3-D mapping and virtual environments," in *Proceedings of Videometrics V (SPIE vol. 3174)*, July 1997, pp. 21–35.

[11] Feng Lu and Evangelos Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–49, Oct. 1997.

[12] Vitor Sequeira, Kia Ng, Erik Wolfart, Joao Goncalves, and David Hogg, "Automated 3D reconstruction of interiors with multiple scan-views," in *Proceedings of Videometrics VI (SPIE vol. 3641)*, Jan. 1999, pp. 106–17.

[13] Gerhard Roth, "Registering two overlapping range images," in *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99)*, Oct. 1999, pp. 191–200.

[14] Robert Bolles and Patrice Horaud, "3DPO: a three-dimensional part orientation system," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 3–26, Fall 1986.

[15] David Eggert, Fitzgibbon, and Robert Fisher, "Simultaneous registration of multiple range views for use in reverse engineering of CAD models," *Computer Vision and Image Understanding*, vol. 69, no. 3, pp. 253–72, Mar. 1998.

[16] Lars Nyland, David K. McAllister, Voicu Popescu, Chris McCue, and Anselmo Lastra, "Interactive exploration of acquired 3d data," in *Proceedings of the 28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*, Oct. 1999, pp. 46–57.

[17] Philippe Robert and Damien Minaud, "Integration of multiple range maps through consistency processing," in *3D Structure from Multiple Images of Large-Scale Environments. European Workshop, SMILE'98*, Reinhard Koch and Luc van Gool, Eds., pp. 253–65. Springer-Verlag, June 1998.

[18] Marc Soucy and Denis Laurendeau, "A general surface approach to the integration of a set of range views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 344–58, Apr. 1995.

[19] James Foley, Andries van Dam, Steven Feiner, and John Hughes, *Computer Grpahics Principles and Practice*, Addison-Wesley, 1987.

[20] Michael Garland and Paul Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of SIGGRAPH 97*, 1997.

[21] Zhengyou Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, Oct. 1994.

[22] Yang Chen and Gerard Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–55, Apr. 1992.

[23] Thomas Cormen, Charles Leiserson, and Ronald Rivest, *Introduction to Algorithms*, MIT Press, 1990.