Automatic 3D Modeling Using Range Images Obtained from Unknown Viewpoints

Daniel F. Huber*
dhuber@ri.cmu.edu
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

In this paper, we present a method for automatically creating a 3D model of a scene from a set of range images obtained from unknown viewpoints. Existing 3D modeling approaches require manual interaction or rely on mechanical methods to estimate the viewpoints. Given a set of range images (views), we use a surface matching system to exhaustively register all pairs of views. The results are verified for consistency, but some incorrect matches may be locally undetectable and correct matches may be missed. We then construct a consistent model from these potentially faulty matches using a global consistency criterion to eliminate incorrect, but locally consistent, matches. The procedure is demonstrated through an application called hand-held modeling, in which a 3D model is automatically created by scanning an object held in a person's hand.

1 Introduction

Recent advances in three dimensional (3D) sensing have led to relatively low cost range imaging devices that accurately measure the 3D structure of a scene from a single viewpoint. Generally, from any given viewpoint, some surfaces in a scene will be unobservable, so data from multiple viewpoints must be combined in order to form a complete model. Current modeling methods require significant manual assistance or rely on mechanical methods to estimate the viewpoints, limiting their applicability in many modeling applications. In this paper, we present a general method to fully automate the 3D modeling process without resorting to these restrictive requirements.

The 3D modeling process involves two main phases: registration, in which the 3D data sets (*views*) are aligned in a common coordinate system; and integration, in which the registered views are combined into a single entity. In this work, we concentrate on the registration phase because this is where the central automation issues lie.

The registration phase can be considered in terms of three interrelated problems: 1) determining which views contain overlapping scene regions (*overlaps*); 2) determining the transform between each pair of overlapping views (*relative poses*); and 3) determining the position of all views in a global coordinate system (*absolute poses*), which is the ultimate goal of the registration phase. Most existing modeling approaches begin with the assumption that approximate relative poses are known. From this, the overlapping views can be determined by applying a suitable definition of overlap to the registered pairs. Johnson [9][10] took the opposite approach, which is more difficult. He assumed that the overlapping views were given, and found relative poses using pair-wise registration with no initial relative pose estimate (*unconstrained pairwise registration*).

Once the overlaps and relative poses are known, the absolute poses can be computed by simultaneously registering all overlapping views (*multi-view registration*). Multi-view registration can be posed as an optimization problem over the continuous space of absolute pose parameters [13]. The objective function seeks to minimize the distance between corresponding points on overlapping surfaces, thereby distributing the errors in the relative pose estimates over the entire model. The overlaps and relative poses provide a good starting point for multi-view registration, which converges to a local minimum.

For automatic modeling, both the overlaps *and* the relative poses are unknown, which makes the registration problem considerably harder due to the mutual dependency between the overlaps and relative poses. We view automatic modeling as the next logical step

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

beyond multi-view registration.

Formally, we define the automatic 3D modeling problem as follows:

> Given an unordered set of range images of a scene *and no additional information*, automatically and robustly construct an accurate 3D model of the scene.

In particular, it is not necessary to know the location of the viewpoints from which the range images were obtained. We denote the N input views $V = \{V_1, \ldots, V_N\}$, where each V_i is initially expressed as a range image R_i, but later as a surface S_i.

The modeling procedure consists of two main phases: surface matching and model construction. In the surface matching phase, the input range images are converted to surface meshes, and a surface matching system [10] performs unconstrained pair-wise registration on all view pairs. The results are verified for consistency, but some incorrect matches may be locally undetectable and correct matches may be missed. In the model construction phase, a globally consistent model is built incrementally from this set of potentially incorrect matches. By using a global measure of consistency, locally consistent, but incorrect, matches can be detected and safely avoided. Once we have a globally consistent model, we perform multi-view registration to find the optimal absolute poses [13], and finally, the registered views are combined using a mesh integration algorithm [5].

Automatic modeling can be posed as a mixed discrete and continuous optimization problem. The discrete optimization is a combinatorial search over the set of models that can be constructed from pair-wise matches, the goal being to distinguish between correct and incorrect matches. The continuous optimization is the same as in the multi-view registration problem. In our current implementation, these two processes are separate, but we are working towards integrating them more closely.

In the remainder of this paper, we begin by summarizing the related work (Section 2). Then we describe three measures of local surface consistency (Section 3) that are used throughout the modeling process. Section 4 presents the details of our modeling procedure, and section 5 illustrates the procedure through an application we call hand-held modeling. Finally, we discuss the algorithm's limitations and our future work in section 6.

2 Related work

Existing methods for constructing 3D models 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 [12]. For smaller objects, absolute poses can be obtained by mounting the sensor on a robot arm [21] or by keeping the sensor fixed and moving the object on a calibrated platform [20]. Relative poses can be estimated by mounting the sensor on a robot equipped only with a relative positioning system such as wheel encoders or inertial sensors [7][11][18].

A common manual registration method is to specify corresponding feature points in pairs of range images, from which relative poses can be estimated [13]. In some systems, corresponding feature points are automatically detected and then manually verified for correctness [7]. Alternately, the 3D data can be aligned directly through an interactive method [15]. In more advanced approaches, a person indicates only which views to register, and performs unconstrained pairwise registration [9][17]. With this approach, the user still must manually verify the registration results.

3 Local surface consistency

Throughout the modeling process, it is often necessary to compare two surfaces to estimate whether or not (or the degree to which) they could represent the same physical surface. We have implemented three local surface consistency measures: overlap distance ($M1$), a general measure that applies to any pair of surfaces; and two measures based on visibility consistency ($M2$ and $M3$), which are tailored to surfaces derived from range images. The measures are defined such that smaller values represent more consistent surfaces, and the input surfaces are assumed to be represented in a common coordinate system (i.e., one view is already transformed by the relative pose). For each measure ($M1$, $M2$, $M3$), we define a corresponding classifier ($CM1$, $CM2$, $CM3$), which is a thresholded version of the corresponding measure. We use these measures in three ways: 1) to rank the results of surface matching; 2) to classify any two surfaces as consistent or inconsistent; and 3) as a basis for a global consistency classifier for verifying entire models.

3.1 Overlap distance

One way to judge the consistency of two surfaces is to directly measure the distance between the surfaces in overlapping regions. We begin with the following definition of overlap:

A point, p, on surface S_i overlaps surface S_j if 1) the point, q, on S_j closest to p is an interior (non-

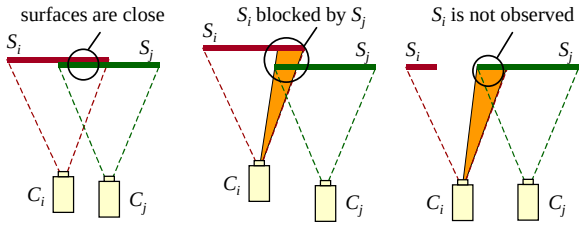

Figure 1: Visibility consistency from the perspective of C_i: an example of correct registration (left), free space violation (center), and occupied space violation (right).

boundary) point of S_j; 2) the angle between the surface normals at p and q is less than a threshold, t_θ; and 3) the Euclidean distance, D, between p and q is less than a threshold, t_D.

Given two surfaces represented as meshes, we can estimate the average overlap distance of surface S_i with respect to S_j:

$$O_D(S_i, S_j) = \frac{\sum_{f \in F_O} w_f A(f)}{\sum_{f \in F_O} A(f)} \quad (1)$$

where w_f is the average of the minimum distance between the corners of face f on S_i and the surface S_j, $A(f)$ is the surface area of f, and F_O is the set of faces on S_i with all three corners overlapping S_j according to the overlap definition above.

We also compute the proportion of S_i that overlaps S_j:

$$O_P(S_i, S_j) = \frac{\sum_{f \in F_O} A(f)}{\sum_{f \in S_i} A(f)} \quad (2)$$

Similarly, O_D and O_P can be computed for S_j with respect to S_i. Since larger overlapping proportions give a more stable estimate of overlap distance, we define our first local consistency measure $M1$ to be the weighted average of the two non-symmetric distances:

$$\overline{O}_D(S_i, S_j) = \frac{O_{P,i,j} O_{D,i,j} + O_{P,j,i} O_{D,j,i}}{O_{P,i,j} + O_{P,j,i}} \quad (M1)$$

where $O_{P,i,j}$ is shorthand for $O_P(S_i, S_j)$ and similarly with $O_{D,i,j}$, $O_{P,j,i}$, and $O_{D,j,i}$.

3.2 Visibility consistency

For surfaces derived from range images, we can develop more powerful measures by looking at the consistency of the two surfaces from the perspective of one of the sensors. For example, consider the surfaces in figure 1 viewed from the sensor position C_i. For a correct registration, the two surfaces have similar

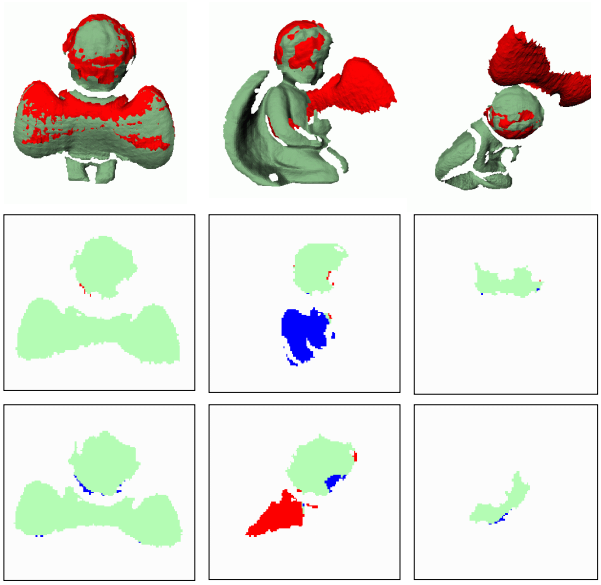

Figure 2: Three match results (top row), with the first view i shown in green/light grey and the second view j shown in red/dark grey, and their corresponding depth differences as seen from the perspective of the first viewpoint $D_{i,j}$ (middle row) and the second viewpoint $D_{j,i}$ (bottom row). In the depth difference images, overlapping pixels are classified according to eq. 4 as same surface points (green/light grey), FSV's (red/medium grey), or "don't care" points (blue/black). For a correct match (left column), most overlapping pixels are same surface points. For an incorrect match (center column), many points will be classified as FSV's from at least one viewpoint (e.g., center bottom). Some incorrect matches (right column) are locally undetectable because the surfaces are consistent from both perspectives.

range wherever they overlap. 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, 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. Free space violations are so named because the blocking surface violates the assumption that the space between the sensor and the sensed surface is empty. Similarly, OSV surfaces violate the assumption that range sensor detects occupied space. Here, we focus on FSV's, but OSV's are discussed further in Section 6. The concept of visibility consistency has been used previously in other 3D vision contexts, including hypothesis verification [2], surface registration [6], range shadow

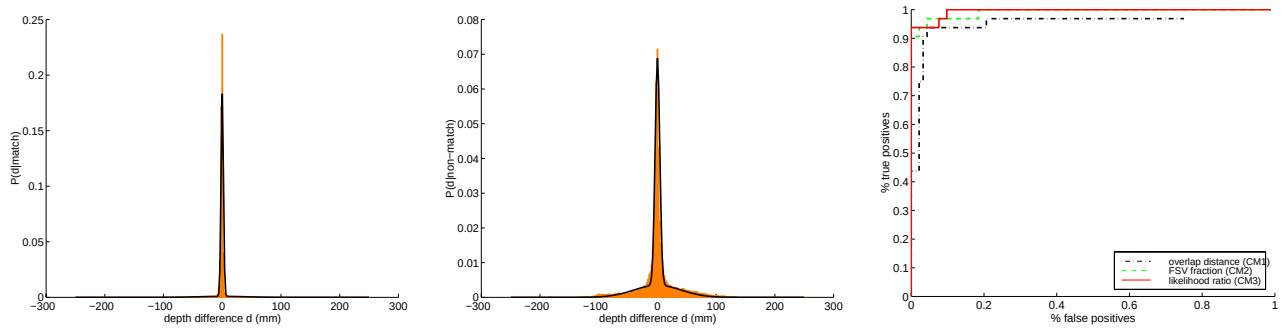

Figure 3: The distribution of depth difference measurements D over a large set of correct matches (left) and incorrect matches (center) from a test object. The predicted distributions, mixtures of two Gaussians learned from a separate training set, are overlaid (thin black line). The ROC curves (right) compare the classification accuracy of the three consistency measures (see text for details).

detection [14], and multi-view integration [16][19].

We can detect FSV's with respect to 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.

To compute FSV's for surfaces S_i and S_j with respect to sensor 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

$$D_{i,j}(k) = Z_j(k) - Z_i(k) \tag{3}$$

is then computed for each pixel $x(k)$ where both z-buffers are defined (Figure 2).

We have developed two measures based on the FSV concept. The first approach applies a threshold, t_{SS}, to the depth difference 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}):

$$x(k) \in \begin{cases} X_{SS}(i,j) & if |D_{i,j}(k)| \leq t_{ss} \\ X_{FSV}(i,j) & if D_{i,j}(k) > t_{ss} \\ X_{DC}(i,j) & if D_{i,j}(k) < -t_{ss} \end{cases} \tag{4}$$

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

$$F(S_i, S_j) = \frac{|X_{FSV}(i,j)|}{|X_{FSV}(i,j)| + |X_{SS}(i,j)|} \tag{5}$$

This computation is independently applied to detect FSV's from the perspective of C_j and combined to form our second consistency measure $M2$:

$$\overline{F}(S_i, S_j) = \max(F(S_i, S_j), F(S_j, S_i)) \tag{M2}$$

Our third local consistency measure 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

$$L(S_i, S_j) = \frac{Pr(H^+|D)}{Pr(H^-|D)} = \frac{Pr(D|H^+)Pr(H^+)}{Pr(D|H^-)Pr(H^-)} \tag{6}$$

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

$$\ln(L(S_i, S_j)) = \sum_{k=1}^{K} \ln Pr(D_{i,j}(k)|H^+) - \sum_{k=1}^{K} \ln Pr(D_{i,j}(k)|H^-) + C \tag{7}$$

where $C = \ln Pr(H^+) - lnPr(H^-)$.

The required probabilities can be estimated from labeled training data. We use a set of matches obtained from exhaustive pair-wise surface matching of the views of a test object. First, we compute separate histograms of the depth differences for the set of correct matches (Figure 3, left) and the set of incorrect matches (Figure 3, center). We then model $P(D|H^+)$ 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 $Pr(H^-|D)$. 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.

Conservatively combining $L(S_i, S_j)$ and $L(S_j, S_i)$ gives our third consistency measure $M3$:

$$\overline{L}(S_i, S_j) = \max(\ln(L(S_i, S_j)), \ln(L(S_j, S_i)) \qquad \text{(M3)}$$

3.3 Comparison of consistency measures

We compare the three consistency measures by evaluating their performance on the task of classifying matches from a test object. By varying the threshold for each classifier ($CM1$, $CM2$, $CM3$) and computing the false positive and false negative rates, we can observe how each measure trades off between the two types of errors. The resulting ROC curves (Figure 3, right) indicate that the two visibility consistency measures are superior to the overlap distance measure. This is because they can detect inconsistencies throughout the sensor's entire viewing frustum.

4 Automatic modeling

We divide the modeling process into two main phases: surface matching and model construction.

4.1 Surface matching phase

In the matching phase, we attempt to register all pairs of views. 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 6).

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 [8].

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 inputs two meshes and outputs a list of relative pose estimates. Details can be found in [9]. 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 don't 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.

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 iterative closest point (ICP) algorithm but extended to handle partially overlapping surfaces [1][22], and a second method that minimizes distances between points and tangent planes in a manner similar to that described by Chen and Medioni [3].

Finally, we perform a local consistency test by applying the classifier $CM3$ to the matches. The classifier threshold is chosen conservatively based on the ROC curve (Figure 3, right) with the intention of eliminating obviously incorrect matches without removing any correct ones.

4.2 Model construction phase

4.2.1 The model graph

The construction and topology of 3D models can be described in terms of an undirected graph $G = (N, E)$, which we call a model graph (Figure 4). A model graph contains a node n_i for each input view V_i. An edge $e_{i,j}$ in G associates V_i and V_j and is annotated with the relative pose $T_{i,j}$ as well as additional information such as the local consistency measures. A connected model graph specifies a *complete model*, since every view can be transformed into a common world coordinate system by compounding relative poses. If, instead, G contains several connected components, each component is 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 pose inconsistencies 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.

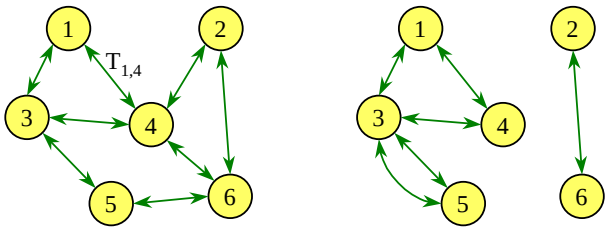

Figure 4: Example model graphs. A complete model (left) and two partial models (right)

4.2.2 Global surface consistency

We are interested in in constructing models that are not only pose consistent but are also globally surface consistent. Global surface consistency is the straightforward extension of local surface consistency to an entire model. A pose consistent model is globally surface consistent if every pair of views is locally surface consistent according to the classifier $CM2$:

$$C_M(G) = \begin{cases} 1 & \text{if} \forall_{(i,j)\in V_C} \overline{F}(S_i, T_{i,j} S_j) \\ & < t_{SS} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

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. We use $CM2$ for computing global consistency instead of the more accurate $CM3$ because the same surface threshold, t_{SS}, can be adjusted to accommodate the accumulation of error from compounding transforms, whereas the distributions used in $CM3$ do not apply to non-adjacent views.

4.2.3 Model construction algorithm

Using the candidate matches from the surface matching phase, we construct an initial model graph G_{SM}. The subgraphs of this graph are the set of all possible model hypotheses for the given matches, each of which may be a complete model or a set of partial models.

The model construction algorithm begins with an empty model graph, G_0, containing the nodes of G_{SM} but no edges, and constructs a consistent model, G, by sequentially adding edges from G_{SM}. The algorithm operates like Kruskal's algorithm [4] for finding the minimum spanning tree of a graph except that at each iteration, an additional check is performed to verify that the model is globally surface consistent:

1: $G \leftarrow G_0$
2: **for all** edges $e_{i,j} \in G_{SM}$, sorted in increasing order using $M3$ **do**
3: **if** n_i and n_j are not connected in G **then**
4: $G' \leftarrow G \cup e_{i,j}$
5: **if** G' is globally surface consistent ($C_M(G')$) **then**
6: $G \leftarrow G'$

Initially, G represents N partial models. Whenever a new edge is successfully added to G, two partial models are joined. Eventually, the algorithm either finds a spanning tree of G_{SM}, resulting in a complete model, or the list of candidate matches is exhausted, resulting in a set of partial models. Step 3 restricts G to be a forest, ensuring pose consistency at each iteration, while step 5 ensures the model's global surface consistency by checking the local surface consistency of all connected pairs.

Figure 5: The hand-held modeling application – a 3D model is automatically constructed from scans of an object held in the user's hand.

Since our model construction algorithm outputs a tree (or a forest), accumulation of errors may lead to visible gaps between overlapping surfaces. Therefore, we apply measure $M1$ to all view pairs to find all overlapping regions and perform multi-view registration [13]. Finally, the surfaces are merged to form a completed model using a mesh integration algorithm [5].

5 Hand-held modeling example

We demonstrate automatic modeling with an application called hand-held modeling, which would not be possible without automatic modeling. For this application, an object is held in the user's hand while range images are obtained from various directions (Figure 5). This is an exceedingly easy method for collecting data, requiring no specialized hardware or training and only about five 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 scanning is complete, the range images are passed to our automatic modeling software, which then produces a 3D model of the object with no manual intervention.

In our experiments, we use a Minolta Vivid 700 laser scanner. A black background and glove allow simple, automatic segmentation of the background by thresholding the intensity image. We scanned 17 range images of a sitting angel (Figure 6). Pair-wise registration resulted in a set of 111 matches (35 correct and

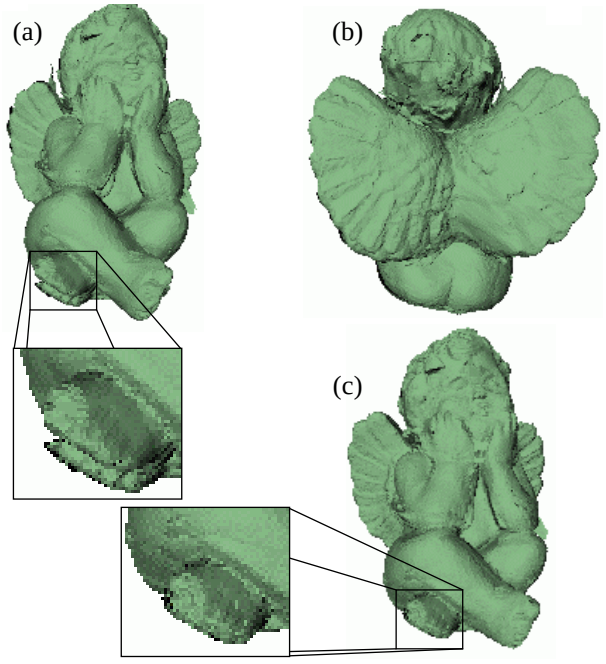

Figure 6: Two views of the output of the model construction algorithm (a-b), and the model after multi-view registration (c)

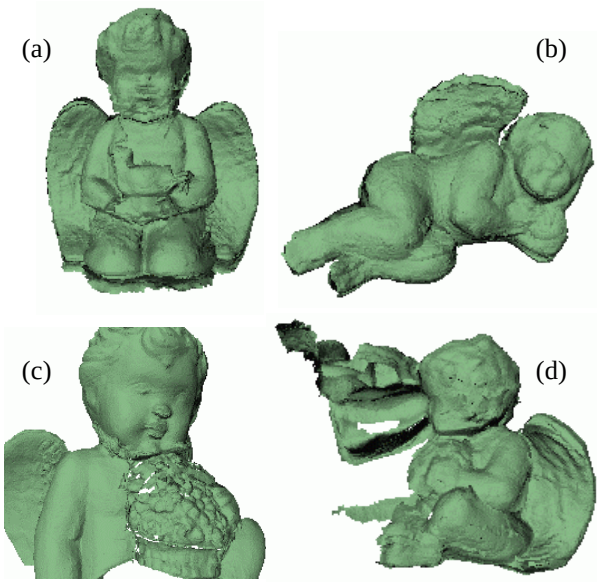

Figure 7: Examples of other correct models (a-c) and an instance where the construction algorithm made a single error (d).

76 incorrect), which was reduced to 44 matches (34 correct and 10 incorrect) by the local consistency test $CM3$. Using these matches, the model construction algorithm produced the model shown in figure 6a-b, which is qualitatively correct. Some error in registration is visible (inset), but multi-view registration eliminates these gaps (Figure 6c. Figure 7 shows several other automatically created models.

6 Discussion and future work

We have identified several aspects of our automatic modeling method to be further developed. Figure 7d 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 (eq. 8). This situation could be avoided with an enhanced test that considered 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).

We can improve the model construction step by integrating the discrete and continuous optimization processes and by using stochastic methods to promote robustness. A trace of the model construction algorithm for the example in figure 7d revealed that the incorrect match was added in the last iteration. A correct match was actually considered earlier in the process but was rejected because the accumulation of relative pose errors made the resulting model inconsistent. This situation can be prevented by performing multi-view registration on the partial models at each iteration of model construction. Still, an incorrect match may be locally more consistent than the best remaining correct match, and our algorithm will choose the incorrect match. This problem can be addressed in two ways. One is to incorporate backtracking into the sequential algorithm, effectively 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_{SM} and then evaluated using the model consistency test. Unfortunately, an enumeration of spanning trees reveals that even for a graph containing mostly correct matches, the chances of randomly selecting a correct spanning tree are extremely small. For the model in figure 7b, the odds are 1 in 64,000 and would require 200,000 independent trials for a 95% chance of success. Our next step is to experiment with other stochastic

methods such as simulated annealing. The current algorithm could be used to generate a starting solution for such methods.

Finally, we must address the issue of view selection. To scale automatic modeling 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.

7 Conclusion

We have presented a method for automatically constructing a 3D model from a set of range images. 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. Throughout the modeling process, we use several local consistency measures, including two tests based on visibility consistency and one based on overlap distance. We demonstrated the practicality of our modeling procedure by automatically constructing 3D models for a number of objects.

References

[1] P. Besl and N. McKay. A method of registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb. 1992.

[2] R. Bolles and P. Horaud. 3DPO: a three-dimensional part orientation system. *International Journal of Robotics Research*, 5(3):3–26, Fall 1986.

[3] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–55, Apr. 1992.

[4] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[5] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH '96*, pages 303–312, 1996.

[6] D. Eggert, Fitzgibbon, and R. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of CAD models. *Computer Vision and Image Understanding*, 69(3):253–72, Mar. 1998.

[7] 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)*, pages 21–35, July 1997.

[8] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of SIGGRAPH 97*, 1997.

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

[10] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–49, May 1999.

[11] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–49, Oct. 1997.

[12] J. R. Miller, O. Amidi, and M. Delouis. Arctic test flights of the CMU autonomous helicopter. In *Proceedings of the Association for Unmanned Vehicle Systems, 26th Annual Symposium*, July 1999.

[13] P. 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*, pages 130–9, Mar. 1997.

[14] L. Nyland, D. K. McAllister, V. Popescu, C. McCue, and A. Lastra. Interactive exploration of acquired 3d data. In *Proceedings of the 28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*, pages 46–57, Oct. 1999.

[15] K. Pulli. Multiview registration for large data sets. In *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99)*, pages 160–8, Oct. 1999.

[16] P. Robert and D. Minaud. Integration of multiple range maps through consistency processing. In R. Koch and L. van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments. European Workshop, SMILE'98*, pages 253–65. Springer-Verlag, June 1998.

[17] G. Roth. Registering two overlapping range images. In *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99)*, pages 191–200, Oct. 1999.

[18] V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan-views. In *Proceedings of Videometrics VI (SPIE vol. 3641)*, pages 106–17, Jan. 1999.

[19] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):344–58, Apr. 1995.

[20] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of SIGGRAPH 94*, pages 311–18, July 1994.

[21] M. Wheeler. *Automatic modeling and localization for object recognition*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, Oct. 1996.

[22] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, Oct. 1994.