Unconstrained Registration of Large 3D Point Sets for Complex Model Building

Owen Carmichael and Martial Hebert
[owenc, hebert]@ri.cmu.edu
The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213

Abstract

We present a method for building models of complex environments from range data gathered at multiple viewpoints. Our approach is unique in that no prior knowledge of the relative positions of the viewpoints is needed in order to register data from them. Furthermore, we present a technique for specification and utilization of so-called "common-sense" constraints on the transformations between views to improve the accuracy and speed of the registration process. Results are shown from our effort to map a 60 m. by 20 m. multiple-room storage area containing a cluttered array of objects.

1.0 Introduction

The problem of building models from multiple views is critical in various applications, including remote operation, virtual environment building, and construction of object libraries for recognition. In this paper, we consider the problem of registering range data sets from multiple locations in order to build a complete model of an environment. A typical scenario involves many viewpoints over a large area with poor or nonexistent initial estimation of the relative positions of the views. Eventually, our goal is to build maps that cover hundreds of square meters at very high (e.g., sub-centimeter) resolution.

This problem has many facets, including computing the transformations between views, correcting the transformations to compensate for drift and error accumulation, merging the views into a single model once the transformations are computed, and incorporating color and texture information into the final model. Here, we focus on the first problem: reliable registration of large point sets from range data. Examples of view merging are shown only to illustrate the results of registration.

Two broad classes of techniques have been developed for point set registration. Methods based on feature matching establish correspondences between point features or surface primitives to compute transformations [3][4]. These approaches do not rely on prior knowledge of the transformations but they do require reliable extraction of the features, which is a difficult problem in itself. Methods based on iterative refinement start with initial estimates of the transformations between views and refine them by iterative minimization of a criterion based on the distance between point sets. The best-known technique of this type is the ICP algorithm originally proposed by Besl and McKay [2]. ICP has been used in model building systems [15] and numerous extensions have been proposed, e.g., by Zhang [17] and Wheeler [16].

Both classes of techniques have severe shortcomings that prevent their application to practical model building problems. First, feature extraction and segmentation is not practical in a general, cluttered environment due to the large number of features and the complexity of the surfaces. Second, the relative positions of views is often not known in advance to a high degree of accuracy; this precludes the use of iterative techniques. For example, position estimates from dead-reckoning of a mobile platform may be too inaccurate for iterative techniques to work, especially over large distances. Also, in some applications, peculiar environmental constraints may prevent the estimation of relative poses. For all these reasons, it is desirable to use a matching technique which is general enough to handle a wide variety of surfaces, and which does not require any prior estimate of the transformation between views.

The method described here is based on a general approach to surface matching [9], which has been applied successfully to object recognition [12] and three-dimensional object modeling. This method has two critical features which address the shortcomings of previous techniques. First, it does not require surface segmentation or feature extraction. Second, it does not require knowledge of the transformation between surfaces prior to registration. Owing to these two features, this surface matching algorithm can be applied to problems of large-scale model building. The goal of this paper is to show how our matching algorithm can be used in the context of mapping large, cluttered environments. Specifically, we use as an example the mapping of a large industrial facility.

This research was supported in part by NSF Grant IRI-97-11853, by an NSF Graduate Fellowship, and by a grant from the Eastman Kodak Corporation.

Although the surface matching algorithm does not require any knowledge of the relative poses of the views, it is not difficult to include pose constraints in the algorithm if they are available. We show that extending the matching algorithm to take advantage of such constraints is straightforward and can reduce the computation time by orders of magnitude, even if the constraints are relatively weak.

2.0 Point Matching Using Spin-Images

In [9], Johnson introduces the spin-image, a two-dimensional descriptor of the local shape of a free-form three-dimensional surface at a point ***p*** on that surface. Spin-images encode the positions of points near ***p*** in terms of distance along and distance from the approximated normal to the surface at ***p***. By comparing this coordinate information from the spin-images of two different points, we arrive at a measure of local shape similarity between the surfaces surrounding those two points.

To construct the spin-image for an arbitrary point ***p***, we first find the best-fit plane to the nearest neighbors of ***p*** and approximate the normal to ***p*** as the normal to this plane. We then define a 2-D basis using the normal ***n*** and the plane $\mathcal{P}$ perpendicular to ***n*** and passing through ***p***. For each point ***x*** in the vicinity of ***p*** we compute its coordinates (α,β) with respect to this basis; α is the distance from ***p*** to ***x*** measured in the plane $\mathcal{P}$, while β is the perpendicular distance from ***x*** to $\mathcal{P}$ (Figure 1.)

The (α,β) values are then discretized and accumulated into a 2-D array of bins called a spin-image; each bin in the spin-image corresponds to some range of α and β values. These spin-images are compact descriptors of the local shape of a surface around a particular point; if two points have similar spin-images, they are considered to have similar local shape (Figure 2).

Spin-image similarity is determined by simple linear correlation of bin values, supplemented with a confidence metric to take into account the number of empty bins in both spin-images. Bins containing no points are not considered in the correlation calculation to help minimize the effects of clutter and occlusion on shape similarity by only considering bins in the spin images that have been filled.

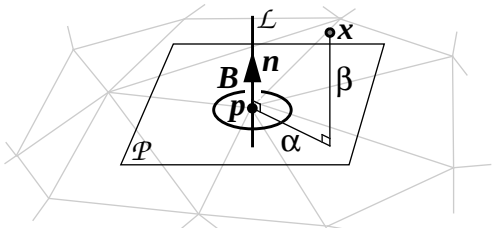

Figure 1: The spin-image for point *p* is found by recording the distance of all nearby points *x* from the surface normal *n* (α) and the distance from *x* to *p* along *n* (β) (from [12]).

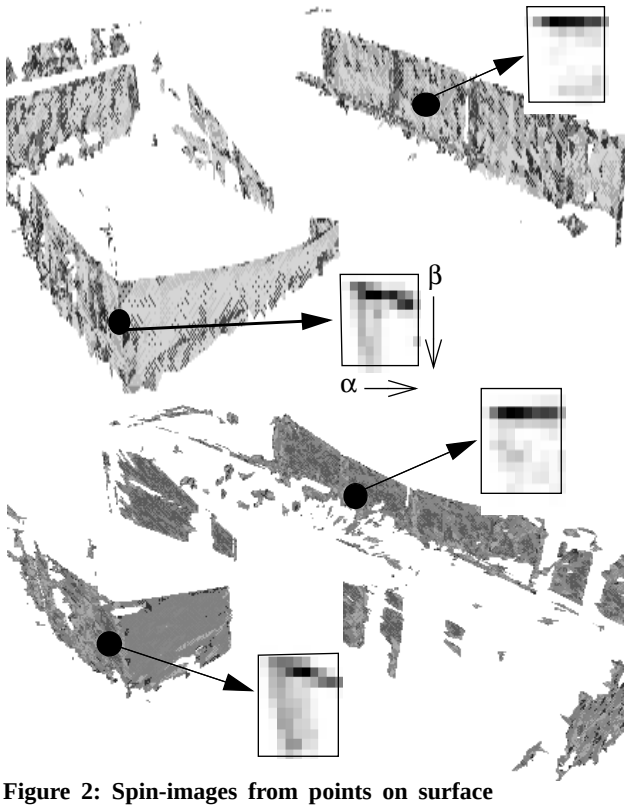

Figure 2: Spin-images from points on surface meshes from two views. Spin-images from corresponding points in the surfaces have similar spin-images.

The final similarity measure between two points takes into account the number of bins used to compute correlation, so that spin-images with the highest amounts of overlap are considered the most similar.

Johnson shows in [9] that through determining the similarity between spin-images of points from two different surfaces, it is possible to recover the transformation that registers the surfaces even though *no initial estimate of this transformation is known at the time of registration.*

3.0 Surface Registration

Spin-image matching provides a powerful tool for building large-scale models from sensor scans taken at unspecified locations. In this section we describe the process of using spin-images to register range data taken from a variety of locations. We assume that regardless of the sensor used, range data from it consists of 3-D points in some fixed coordinate frame. Given these points, a triangular mesh is formed by connecting nearest neighbors, and noisy points and edges in this mesh are removed through cleaning and smoothing operations. At this point, we have a high-resolution 3-D representation of the model as seen from a particular viewpoint; for reasons to be explained below, we then simplify this surface using the algorithm

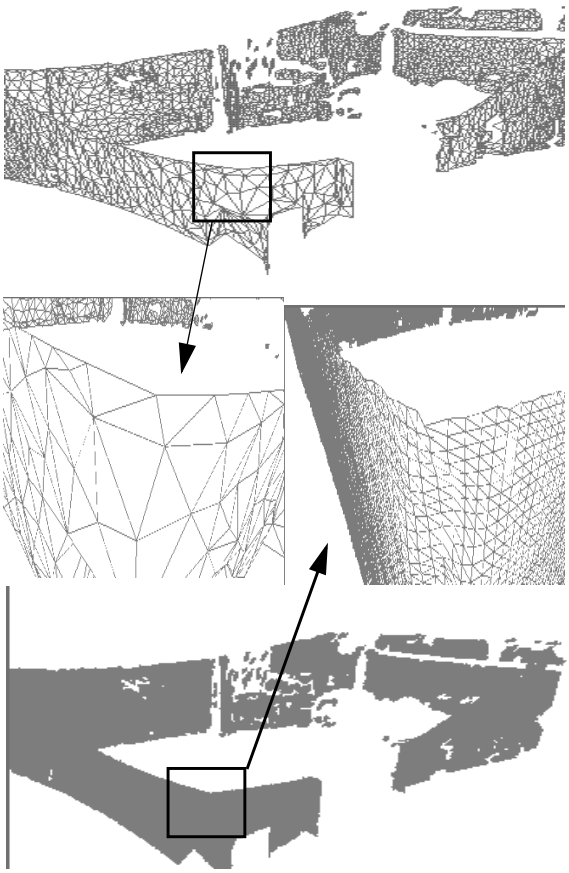

Figure 3: A view of the interior, shown at low resolution (top) and high resolution (bottom), with detail magnified to show the difference in resolution.

found in [11] to obtain a low-resolution version of the surface, as shown in Figure 3.

Given two low-resolution meshes, we register them using spin-image matching as follows. First, a fixed fraction of points is selected at random from both surfaces. Spin-images are produced for these points, and all the spin-images from one surface are compared to all the spin-images from the other using the similarity measure described above. When a pair of spin-images are found to have high similarity, the points that produced them are considered to correspond to each other; after all spin-image comparisons have taken place, we are left with a set of point matches between the two surfaces. Finally, an estimate of the transformation that aligns the two surfaces is computed from the point matches using the algorithm of [5].

The transformation aligning corresponding points is considered only a rough estimate of the true transformation between the range data sets for two reasons: first, the low-resolution mesh from which correspondences are drawn is neccessarily an approximation of the raw data, and second, the set of correspondences that determines the transformation consists of only a subset of the points on the simplified surface. For these reasons, after we find the transformation that aligns the low-resolution meshes we apply it to the high-resolution meshes; we then use the ICP algorithm [2] on the roughly-aligned high-density surfaces to refine the transformation estimate. The resulting transformation aligns the full range data sets with acceptable accuracy.

Registering low-resolution meshes and refining the transformation estimate on high-resolution meshes was found in practice to be much more efficient than registering the high-resolution meshes directly. For example, one full-resolution data set like the one seen in Figure 2 contains roughly 65000 points (compared to roughly 4000 points in the adjacent low-resolution mesh); registering two meshes of this size is computationally infeasible due to the number of spin-image comparisons required. In many cases, the transformation that aligns the low-resolution surfaces aligns the high-resolution surfaces so well that the ICP procedure does not need to do much work to reduce the error between the surfaces below an acceptable level.

Once several surfaces have been registered together, they are transformed into the same coordinate frame and merged using the algorithm found in [10]. The resulting mesh incorporates range data taken from any number of viewpoints, and thus may include sets of points which could not have been visible simultaneously from any single viewpoint. Through merging several registered high-resolution surfaces, we build up our final model.

Two features of this procedure make it convenient for users wishing to build large, complex models. First, the only step in the process which depends on the user's choice of range sensor is the acquisition of range data; the manipulation of meshes, registration of surfaces, ICP, and merging are completely independent of the range sensor used. Furthermore, since no estimate of the transformation between views is required for the registration procedure to return the correct result, the user is allowed to move the sensor freely between snapshots without needing to precisely measure the difference in position and orientation between successive locations. These two features make registration of range data from a large number of viewpoints fast and largely automatic.

4.0 Detailed Example

This method of surface registration has been successfully applied to build a model of a large warehouse space composed of two adjacent rooms. The building measured

roughly 60 meters long by 20 meters wide by 10 meters high, and was filled with an assortment of clutter and debris as shown in Figure 4. We used a K2T/ZNF laser

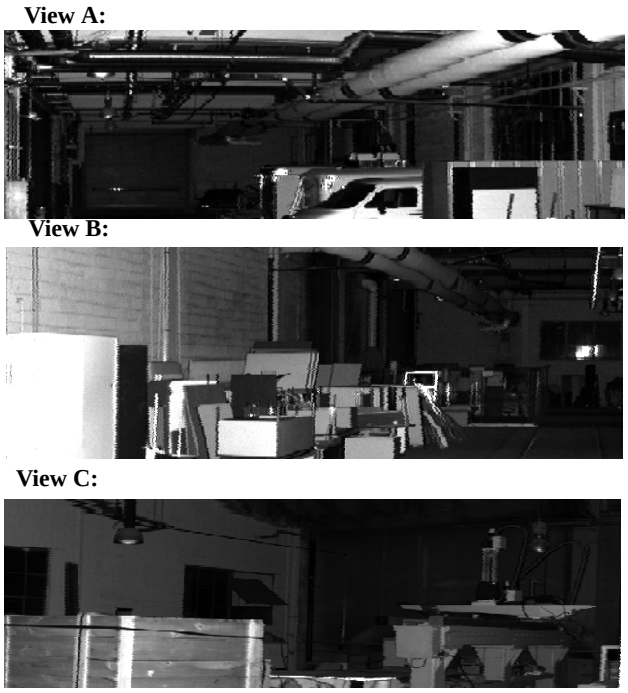

Figure 4: Three reflectance images of different sections of the warehouse area; view A (top), view B (middle), and view C (bottom). Only part of the panoramic scan is shown for each image.

range finder [6][7] mounted atop a minivan to collect range data at 32 different points in the warehouse; each scan of the sensor covered a 360-degree field of view with 20 degree depression and a maximum range of about 40 meters. The resolution of the sensor is rated in the millimeter region.

Each scan of the range finder returned 1.8 million 3-D points; through naive subsampling this set was reduced to 65000 points. The resulting point cloud was converted into a mesh as described in Section 2 and resampled to roughly 5000 points for registration.

Surface meshes produced from each of the 32 data sets were successfully registered to the meshes of adjacent scans. Figure 5 shows the registration results for two of the meshes. With the transformations between neighboring meshes known, it was then possible to align all meshes in a common coordinate system so that they could be unified into a single mesh that covered the entire warehouse space. Because of limitations in our implementation of the merging algorithm, only 20 of the views were merged into the final model, although all 32 data sets were registered. The

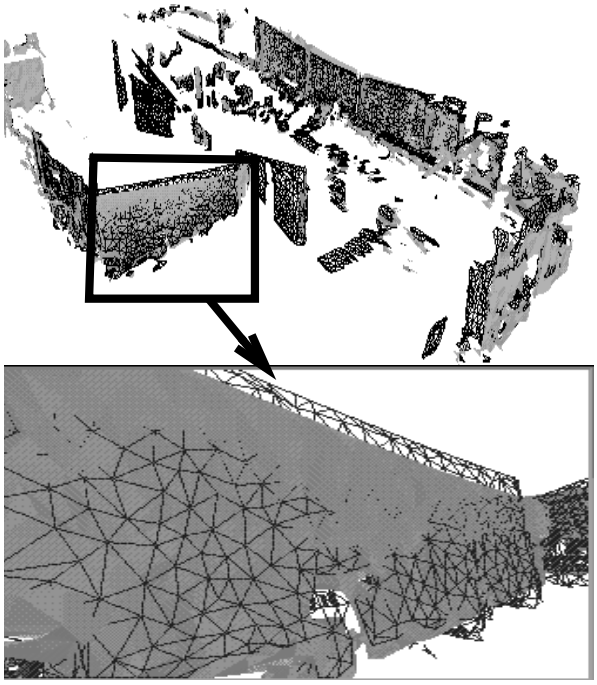

Figure 5: (Top) Registered range data acquired from two separate sensor locations. One data set is displayed in black wireframe, the other in solid grey. (Bottom) Boxed section enlarged to show detail.

complete mesh of the warehouse contained 138000 points at full resolution, while a low-resolution version of the model contained 25000 points. The resolution of the final mesh was low compared to the that of the constituent high-resolution views due to limitations of the mesh merging implementation; otherwise, it would have been possible to produce a final mesh with no resolution loss. Statistics for each of the different types of surface mesh-- high-resolution single-viewpoint, low-resolution single-viewpoint, high-resolution complete, and low-resolution complete-- are summarized in Table 1. Resolution is measured as the average length of edges in the mesh, in millimeters.

TABLE 1. Mesh Statistics

Mesh Type	# Points	# Edges	Resolution
High Res.	65000	160000	70
Low Res.	5000	12000	275
Complete High Res.	138000	400000	150
Complete Low Res.	25000	65000	320

Figure 6 shows a bird's-eye view of the final model, while Figure 7 shows the positions of the sensor at viewpoints A, B, and C from Figure 4 along with views of the final model from these vantage points. As mentioned

above, we did not attempt to refine the final merged model; more sophisticated merging algorithms can certainly be used once the transformations are computed.

Figure 6: Bird's-eye view of the building model.

5.0 Transformation Clipping

While it is important that no a priori estimate of the transformation between two surfaces is required for the algorithm described in Section 3 to successfully register them, a rough estimate of the transformation can be helpful in guiding the search for possible point correspondences and safeguarding against registration mistakes. In our procedure for surface registration, we allow the user to specify information about the ranges of possible translation and rotation between viewpoints to improve registration performance; we refer to the truncation of the space of allowed transformations as "transformation clipping."

Let A and B be the two surfaces to be registered. In the original spin-image matching algorithm, a fixed percentage of points is picked at random from both A and B as representative samples of the surfaces. To search for point correspondences, each point from A's sample is compared to each point in B's, regardless of the relative positions of the points on their respective surfaces. This approach encourages comparisons between points on the two surfaces that clearly don't correspond to each other; two difficulties in the registration process emerge.

First, if a section of surface A is locally similar enough to a non-corresponding section of surface B, a wildly inaccurate registration can result; in this way, the user's common sense about the relative locations of the two surfaces is not enforced. For example, Figure 8 shows a scenario in which a range sensor is mounted on a wheeled mobile robot in a hallway with corners. When registering the first range mesh (shown in black) with the second (shown in grey), it appears, purely on account of the shape of the meshes, that the best alignment between them is shown in the righthand part of the figure. However, the transformation that registers the meshes in this way aligns the ceiling from the first data set with the floor of the second and vice versa. Clearly

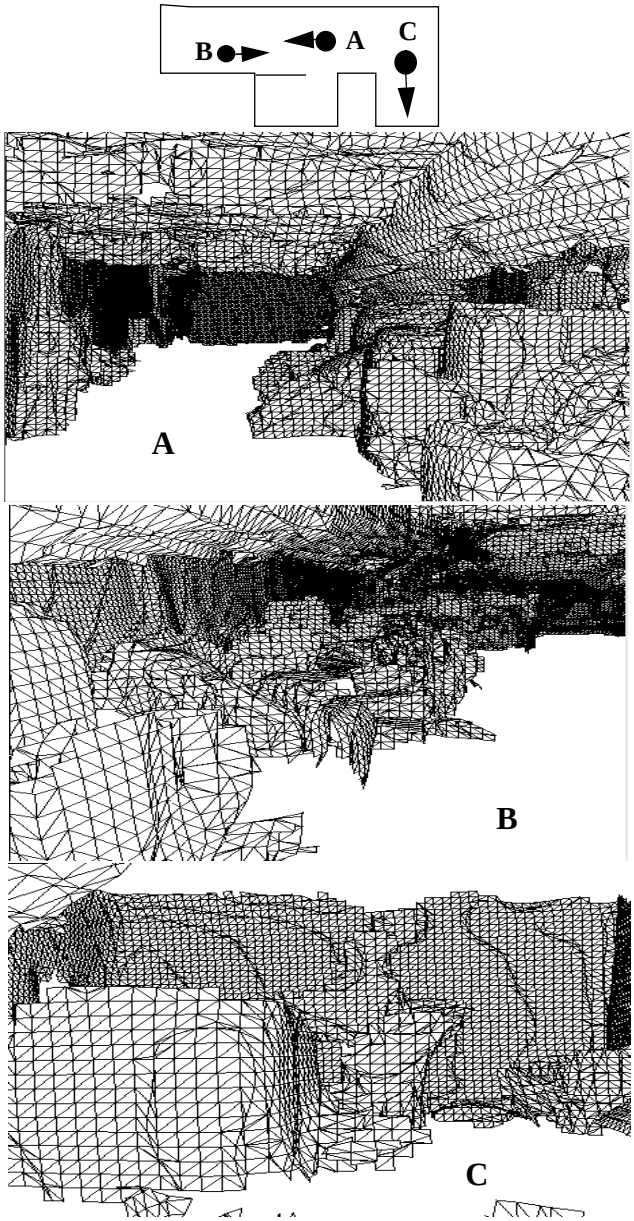

Figure 7: Views of the final model from viewpoints A, B, and C of Figure 4.

this interpretation of the data is unacceptable, given the setup of the sensor and our assumptions about how the robot can move (i.e., that this particular robot cannot turn end-over-end between shots). Although this particular mistake did not occur in the warehouse example shown in the previous section, it is clear that it can occur whenever the matching between data and map is ambiguous. Regardless of whether or not the sensor is mounted on a mobile robot, there are situations in which we wish to leverage our common sense about the transformation between data sets in order to safeguard against unrealistic interpretations of the data. For example, the range sensor

may be mounted on a tripod which is moved about in a room with a flat floor; in this case we know that the sensor's elevation will remain the same at each shot, give or take small changes due to a slight slope in the floor. In other scenarios, the sensor may be mounted on a track which allows it to translate in only one direction, or it may always be aimed at the walls rather than towards the ceiling or floor. Each of these situations implies a truncation of the transformation space; incorporating this knowledge into the registration process is desirable to enforce our intuitions about how the data was gathered.

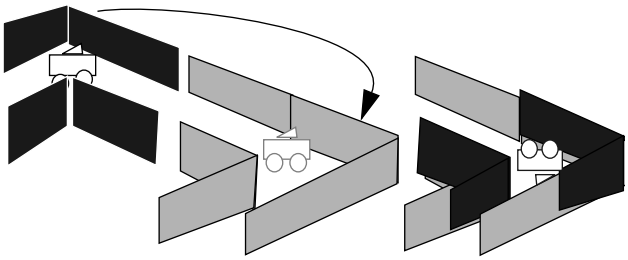

Figure 8: The robot takes range data from two different positions in the hallway (left). The registration which best aligns the two data sets flips the black mesh 180 degrees and implies the unlikely situation that the robot flipped onto its head between scans (right). This situation could be easily avoided by using common-sense geometric constraints.

The second reason to enforce transformation constraints involves speed of execution. If n is the number of points in the point sample for a particular mesh, then assuming each mesh selects the same number of points the process of comparing spin images takes $O(n^2)$ pointwise comparisons; for large surface meshes, this process can be prohibitively expensive so any reduction in the correspondence search space will reduce execution time.

We therefore allow the user to specify minimum and maximum possible values for the three translational and three rotational components of the transformations between meshes. These transformation constraints are used to compute, for each point in each mesh, its maximum and minimum possible displacement along the three dimensions; these displacement restrictions are used to decide which spin-images from the two meshes should be compared to each other. As shown in Figure 11, this technique defines a bounding box of motion around each point; thus the transformation constraints are not enforced especially tightly. However, we demonstrate in Figure 12 that the number of spin-image comparisons nonetheless drops dramatically as a result of limiting the correspondence space in this way. The bounding-box approach to transformation clipping easily extends to cases in which the set of possible transformations consists of multiple subspaces of the transformation space; in other words, the method could readily be applied to scenarios where we know that between scans the sensor translated along and rotated about a particular axis *or* rotated about a different axis.

Given a point in a coordinate frame *X, Y, Z*, we describe a 3-D transformation to this point as a rotation by ω_x about X, followed by a rotation of ω_y about Y, followed by a rotation of ω_z about Z, followed by translations t_x, t_y, and t_z along the *X, Y,* and *Z* axes (this is the *X-Y-Z* fixed angle convention of [8]). Let $y_x(\mathbf{x},\omega)$ and $z_x(\mathbf{x},\omega)$ be the y and z coordinates of a point $\mathbf{x} = (x,y,z)$ after it has been rotated about *X* by ω; similarly $x_y(\mathbf{x},\omega)$ and $z_y(\mathbf{x},\omega)$ are the x and z coordinates of a point **x** after it has been rotated by ω about *Y* and $x_z(\mathbf{x},\omega)$ and $y_z(\mathbf{x},\omega)$ are the x and z coordinates of **x** after it has been rotated about *Z* by ω. With these notations:

$$z_x(x,\omega) = y\sin\omega + z\cos\omega \qquad z_y(x,\omega) = z\cos\omega - x\sin\omega$$
$$y_x(x,\omega) = y\cos\omega - z\sin\omega \qquad y_z(x,\omega) = x\sin\omega + y\cos\omega$$
$$x_y(x,\omega) = z\sin\omega + x\cos\omega \qquad x_z(x,\omega) = x\cos\omega - y\sin\omega$$

We are given constraints on the ranges of the rotations and translations allowed to align surface A to surface B, specifically maximum and minimum rotations ω_{xmin} and ω_{xmax} about the *X* axis, minimum and maximum translations t_{zmin} and t_{zmax} along the *Z* axis, and so on. Looking at $\mathbf{x}_{yz}$, the projection of **x** into the *Y-Z* plane, we can easily compute the maximum value that $z_x(\mathbf{x},\omega)$ may assume given **x** and given that $\omega_{xmin} < \omega < \omega_{xmax}$ by observing that one of two possible cases occur as shown in Figure 9; either the rotation angle that would align $\mathbf{x}_{yz}$ with the positive *Z* axis is between ω_{xmin} and ω_{xmax}, or it is not. In the first case, rotating **x** so that $\mathbf{x}_{yz}$ lines up with the positive *Z* axis maximizes $z_x(\mathbf{x},\omega)$; otherwise, rotating by either ω_{xmin} or ω_{xmax} maximizes $z_x(\mathbf{x},\omega)$. Similarly, we can find the maximum of $y_x(\mathbf{x},\omega)$ by checking whether the angle that aligns $\mathbf{x}_{yz}$ to the positive *Y* axis is between ω_{xmin} and ω_{xmax}; if so, then rotating **x** by this angle maximizes $y_x(\mathbf{x},\omega)$, and if not, then rotating by either ω_{xmin} or ω_{xmax} maximizes $y_x(\mathbf{x},\omega)$. Furthermore, the minimum values of $y_x(\mathbf{x},\omega)$ and $z_x(\mathbf{x},\omega)$ may be found similarly by examining whether the interval $[\omega_{xmin},\omega_{xmax}]$ contains the angles that would align $\mathbf{x}_{yz}$ with the negative y and z axes, respectively.

Let z_{xmax} and y_{xmax} be the maxima of $z_x(\mathbf{x},\omega)$ and $y_x(\mathbf{x},\omega)$ and let $\mathbf{x'} = (x,y_{xmax},z_{xmax})$. By computing the maxima and minima of $x_y(\mathbf{x'},\omega)$ and $z_y(\mathbf{x'},\omega)$, we find the maximum and minimum possible *X* and *Z* values that the point **x** can take on after it is rotated about the *X* and *Y* axes by angles in the appropriate ranges. Extending this analysis to rotation about the *Z* axis and translations along *X, Y,* and *Z*, we arrive at maximum and minimum *X, Y,* and *Z* values for the point **x**.

Thus, for each point $\mathbf{x_A}$ from surface A, we compute the

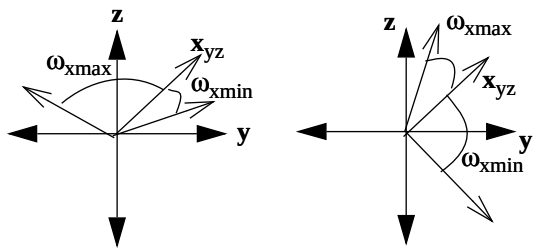

Figure 9: The two cases to consider when determining the maximum attainable z value under rotations between ω_{xmin} and ω_{xmax}: either it's possible to align x_{yz} with the positive z axis (left) or it is not (right).

maximum and minimum possible *X, Y,* and *Z* values for $\mathbf{x_A}$ under transformations in the ranges supplied by the user. For each point $\mathbf{x_B}$ from surface B, If $\mathbf{x_B}$'s coordinates fall between these maxima and minima, then we compare the spin images for $\mathbf{x_A}$ and $\mathbf{x_B}$ for possible correspondence; otherwise, we know that $\mathbf{x_A}$ could not possibly be transformed to $\mathbf{x_B}$ according to the constraints specified by the user, so the two points are not considered candidates for a correspondence.

An example of two scenes registered with transformation clipping is shown in Figure 10 and Figure 11. The range

Figure 10: Surface mesh for one view of the building with one point of the mesh highlighted in black.

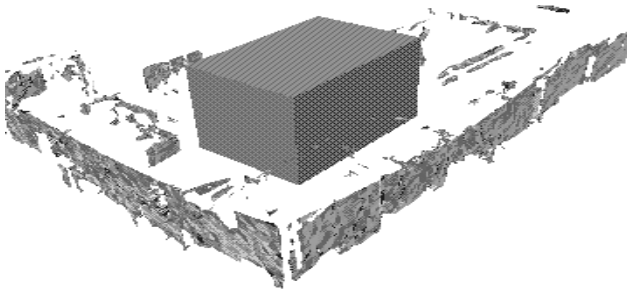

Figure 11: Mesh for a second view of the interior. Only those points inside the grey box will be compared to the highlighted point in Figure 10.

data for the first snapshot was taken with the sensor positioned at the point marked; it was then moved forward roughly 1.5 meters and the data for the second snapshot was taken. When registering the two surfaces, we estimated that the sensor translated forward between 1 and 2 meters, and to account for any inaccuracies in motion, we estimated that the sensor could have translated to the left or right up to 100 cm, translated up or down up to 100 cm, and rotated up to 5 degrees in any direction. (Upon registering the two surfaces, we found that the sensor had actually moved forward roughly 1.4 m, up 15 cm, and to the right 33 cm; rotations were all less than 1 degree). Figure 12 shows the average number of spin-image comparisons needed per point when registering the two surfaces at different resolutions, with and without transformation clipping in use.

6.0 Performance and Limitations

6.1 Accuracy

One way to assess the accuracy of the registration procedure is to examine the closest-point error between the high-resolution meshes after the ICP procedure. The closest-point error for a particular point in a mesh is the euclidean distance between that point and the closest point to it in the mesh it has been registered to; in the warehouse example, we found that the average closest point error was on the order of 1 cm after ICP had been run on the high-resolution mesh. We also ran ICP on the low-resolution meshes after they had been aligned using the rough transformation estimate, and found that the average closest-point error was roughly 15 cm, thus exhibiting the utility of high-resolution transformation refinement.

Another measure of registration accuracy is simply the number of correspondences found between meshes. On average, about 1000 correspondences were found during the registration process; in other words about 20% of the points from each mesh were found to match each other. With such a significant fraction of range points contributing to the transformation estimate, we can be relatively confident in its fidelity.

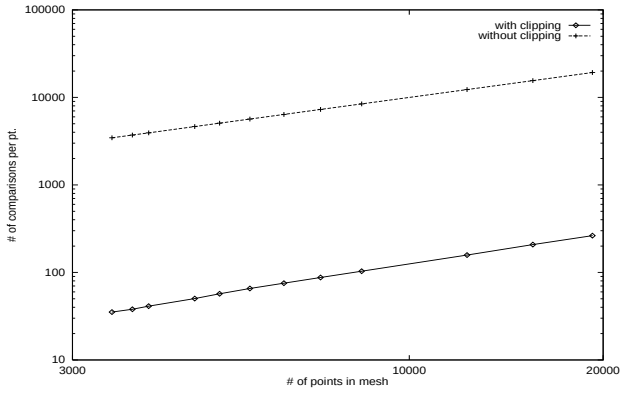

Figure 12: A plot of the average number of spin-image comparisons needed per point with and without transformation clipping, as a function of point density of the mesh.

6.2 Limitations

It is apparent from the overhead view of the final mesh of the warehouse (Figure 6) that the model of the building drifts to the right along its length, whereas in reality none of the walls of the building are curved. This error is introduced by the order in which surfaces are registered to each other and transformed into a common coordinate system. Each mesh is only registered to the mesh produced from the next nearest sensor scan; the recovered transformations are then daisy-chained together to transform all the meshes into a single coordinate system. As a result, slight errors in any transformation are propagated onto the next transformation in the chain; the error accumulates to the point that the transformation that shifts the last mesh in the chain into the common coordinate system may be noticeably inaccurate. Also, while it is clear that two meshes must overlap to a certain degree in order for registration to be successful, it is difficult to predict what the system will do if the meshes do not. Generally speaking, a better method for evaluating the quality of the match is needed.

7.0 Conclusion

Several enhancements would help to make certain aspects of this process more automatic and user-friendly.

Resolution of the simplified meshes used to register individual views is selected manually by the user. Ideally, we would prefer that the user not need to know about such details; we are hoping to develop methods for automatic selection of mesh resolution based on the shape complexity of the surfaces.

Another decision we wish to free the user from concerns the design of spin-images. Currently, the user decides how many bins the spin-images should have in the α and β directions (its "width" and "height") and the size of the volume of space covered by each bin in the spin-image (its "resolution"). Future work will address the problem of automatically determining what values these parameters should take to ensure that the spin-images for a particular mesh express its shape characteristics most descriptively so that the ambiguities in the registration process will be minimized.

Also, while our final model of the garage interior is a representation of the shape of the surfaces found in the building, it is not a complete depiction of the real scene in that no reflectance or color information is incorporated into the model. Another avenue for future research is the simultaneous registration of range and reflectance data, as well as the simultaneous merging of range images and co-registered intensity maps.

References

[1]R. Bergevin, D. Laurendeau and D. Poussart, Registering range views of multipart objects, Computer Vision and Image Understanding, vol. 61, no. 1, pp. 1-16, 1995.

[2]P. Besl and N. McKay. A method of registration of 3-D shapes. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 2, pp. 239-256, February 1992.

[3]Y. Chen and G. Medioni. Surface description of complex objects from range images. Proc. IEEE Computer Vision and Pattern Recognition (CVPR '94), pp. 153-158, 1994.

[4]C. Chua and R. Jarvis. 3-D free-form surface registration and object recognition. Int'l Jour. Computer Vision, vol. 17, no. 1, pp. 77-99, 1996.

[5]O. D. Faugeras and Martial Hebert. The representation, recognition, and locating of 3D shapes from range data. International Journal of Robotics Research, 5(3):27-52, 1986.

[6]Froelich, C., M. Mettenleiter, F. Haertl. Imaging laser radar (LIDAR) for high-speed monitoring of the environment. *SPIE Proceedings of the Intelligent Transportation Systems Conference*,October 1997.

[7]Hancock, J., E. Hoffman, R. Sullivan, D. Ingimarson, D. Langer, M. Hebert. High-performance laser range scanner. *SPIE Proceedings of the Intelligent Transportation Systems Conference*, October 1997.

[8] J. Craig. Introduction to robotics: mechanics and control. Addison-Wesley Publishing Co., Readng, MA.

[9]A.Johnson. Spin-Images: A Representation for 3-D Surface Matching. PhD Thesis, The Robotics Institute, Carnegie Mellon University, August 13th, 1997.

[10] A. Johnson and S.B. Kang. Registration and integration of textured 3-D data. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97), pp. 234-241, Ottawa, Ontario, May 12-15, 1997.

[11] A. Johnson and M. Hebert. "Control of Polygonal Mesh Resolution for 3-D Computer Vision." Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-96-20, February, 1997.

[12]A. Johnson and M. Hebert. Object recognition by matching oriented points. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[13]H.Y.Shum, R. Reddy, M. Hebert, K. Ikeuchi. An integral approach to free-form object modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence. December 1997.

[14]M. Soucy and D. Laurendeau. Multi-resolution surface modeling from multiple range views. Proc. IEEE Computer Vision and Pattern Recognition (CVPR '92), pp. 348-353, 1992.

[15]G. Turk and M. Levoy. Zippered polygonal meshes from range images. Proc. Computer Graphics (SIGGRAPH '94), pp. 311-318, 1994.

[16]M. Wheeler. Automatic Modeling and Localization for Object Recogntion. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, October 1996.

[17]Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. Int'l Jour. Computer Vision, vol. 13, no. 2, pp. 119-152, 1994.