Range Image Registration: A Software Platform and Empirical Evaluation

Gerald Dalley and Patrick Flynn
Signal Analysis and Machine Perception Laboratory, The Ohio State University
email: {dalleyg, flynn}@ee.eng.ohio-state.edu

Abstract

Building 3D models of real-world objects by assembling views taken by a range sensor promises to be a more efficient method than manually producing CAD drawings. In this technique, a series of range images are acquired and then registered or aligned with each other to a high degree of accuracy. Finally, the polygonal meshes corresponding to the range images are merged to form a complete 3D model consisting of a single mesh.

Many techniques have been proposed to solve the registration problem; however, little work has been done to date to compare several registration algorithms with the same sets of data. In this paper, we examine a software test-bed built for performing such comparisons. Within this test-bed, we have implemented several common registration algorithm variants to the baseline Iterative Closest Point (ICP) algorithm and tested them on partially overlapping range images taken from four different objects.

1. Introduction

In recent years, there has been growing interest in techniques for building 3D computer models of real-world objects and scenes without requiring humans to manually produce these models using laborious and error-prone CAD-based approaches. Using range sensors, users are able to capture 3D images of objects from different viewpoints that may be combined to form the final model of the object or scene [3]. These models then may be used for a variety of purposes such as building 3D maps for robot navigation, providing training data for computer vision experiments, and digitizing historical buildings for restoration planning [16], [18].

The first model-building step is to obtain a set of range images. A variety of techniques exist for dense range image acquisition, with time-of-flight, structured light, and laser triangulation among the most popular approaches [1].

The range images must next be registered (placed in a common coordinate system). This is essentially an alignment process. Requiring humans to perform these very precise alignments is impractical. Additionally, while some sensors provide positional and orientation data for each range image, these values are generally too inaccurate for the purposes of model building. Even though both of these alignment methods are imprecise, either may be used to generate a coarse registration that may then be refined through a suitable automatic fine registration scheme. Beginning in Section 2, we will discuss fine registration in more detail.

After alignment, the data are combined to produce a single surface description. This mesh integration step prunes the redundant data from the input range images and stitches together their surfaces. Various techniques have been devised to perform this task such as Turk's point-based zippering [17], several volumetric space-carving techniques [5], [12], and converting to curve-based representations [9], [10]. Mesh integration is beyond the scope of this paper.

The remainder of this document is organized as follows. Section 2 describes the most common methods used for automatic range image registration and the principal approaches that researchers have taken to solve the problem. In Section 3, we describe the rationale behind our work in building a software test-bed to comparatively study various range image registration approaches. An overview of the test-bed implementation follows in Section 4, and the results and conclusions of comparing several registration methods using this software are given in Section 5.

2. Prior Work

Early work in the field generally concentrated on extracting and aligning major features of the range images. These techniques were generally used for recognition of pre-built models instead of model construction [3]. Some researchers have more recently had success in adapting feature-based techniques for the building of models [4], [16]. Besl and McKay introduced another approach, the *Iterative Closest Point* (ICP) method in [2]. This method iteratively minimizes the distance between the range images by finding corresponding point pairs. A significant number of researchers use this algorithm in one form or another [16]. Related to the ICP approach is one that matches surface normals with tangent planes [3]. A third group of researchers use a "spring-mass" based system which simulates a series of damped oscillations on corresponding point pairs based on spring-mass physics systems [7], [18]. In this chapter, we will describe in more detail the non-feature-based approaches. In particular, we will discuss the ICP algorithm and its variants in the greatest depth.

If a range image, P, is being registered to a data set X by using a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{T}$, then the ICP registration process attempts to minimize the objective function

$$f(\mathbf{R},\mathbf{T}) = \frac{1}{N_P}\sum_{i=1}^{N_P}\left\|\boldsymbol{x_i} - \mathbf{R}\boldsymbol{p_i} - \mathbf{T}\right\|^2$$

where $P=\{\boldsymbol{p_i}\}$ is the set of points in the range image being registered, $X=\{\boldsymbol{x_i}\}$ is the set of points in the reference range image to which P is being registered (with $\boldsymbol{x_i}$ the closest point in data set X to the point $\boldsymbol{p_i}$) N_p is the number of points in data set P, $\mathbf{R}$ is the 3x3 registration rotation matrix, and $\mathbf{T}$ is the registration translation vector. Besl uses a quaternion-based approach to find the values for $\mathbf{R}$ and $\mathbf{T}$ that minimize this function [2]. These calculations are performed iteratively using the following basic algorithm:

1. Given each point $\boldsymbol{p_i}$, find the closest point, $\boldsymbol{x_i}$, in X to $\boldsymbol{p_i}$.
2. Compute the registration ($\mathbf{R}$, $\mathbf{T}$) such that $f(\mathbf{R}, \mathbf{T})$ is minimized.
3. Apply the registration to P.
4. Go to step 1 if the difference in the registration errors, $|f_n(\mathbf{R}, \mathbf{T}) - f_{n-1}(\mathbf{R}, \mathbf{T})|$, has not dropped below some threshold.

This process is guaranteed to converge to some local minimum for any starting registration when P is a subset of X.

The basic ICP algorithm has been modified in many ways by various researchers in attempts to improve the speed and/or quality of the registrations it produces. For example, in [2], Besl and McKay recommend accelerating the iterative process by using a parabolic interpolant to the three previous iterations.

Turk and Levoy [16] enhance the registration by adding a confidence-weighting factor to each point pair. Points whose normal is directed away from the range sensor are given a lower confidence value, as are points at the edge of the mesh. These confidence values are used to weigh the summed terms of the minimization function and enhance the ICP algorithm to be more robust to sensor errors.

The original algorithm requires that a range image be registered to a surface that is a superset of the range image. Unfortunately, when building new models from range images, no model to which a range image may register exists. Instead, we desire to register multiple range images so that their *overlapping* regions are aligned. Schütz *et al.* [15] propose a simple heuristic method of determining which corresponding point pairs belong to overlapping regions and which are actually not corresponding points. They theorize that point pairs whose distance is much greater than the separation of the centers of mass of two partially registered range images must be outliers. They calculate a binary weighting factor for each point pair as follows:

$$w_i = \begin{cases} 1 & \text{if } f_n(\mathbf{R},\mathbf{T}) < (c \cdot s \cdot r)^2 \\ 0 & \text{otherwise} \end{cases}$$

where s is the range scanner sampling distance, r is the subsampling factor, and c is the empiricially deteremined threshold based on the separation of the centers of mass of the data sets. Those pairs whose two points are separated by more than a specified value d are considered outliers and given a weight of zero. Those point pairs whose distance is less than d are considered inliers and have a weight of unity. In this way, those points that are highly separated and likely to belong to non-overlapping regions are excluded from the registration calculations. In addition, Schütz *et al.* introduce a "surface coupling" measure ε. This measure indicates the percentage of points in a data set that are counted as contributing to the overlapping region. They suggest that when ε drops below 30 to 50%, the validity of the ICP step is in question. Thus a program can detect when it has begun excluding too many point pairs. Schütz *et al.* further expand the ICP algorithm by enhancing the convergence decision process to include stability of interpoint distance variation.

Zhang has implemented a more extensive and sophisticated set of modifications to ICP for the purposes of robot navigation [18]. In addition to aligning both points and curves, a more theoretically-based method to classify outlier corresponding point pairs is used. Central to this method is the assumption that when the mean point pair distance μ is on the order of the range image sampling distance s, the distances between points in the corresponding point pairs will follow a Gaussian density distribution. Given a coarse registration of two range images with significant overlap, a point $\boldsymbol{x_i}$ not belonging to the overlapping region will have large distances to its corresponding $\boldsymbol{p_i}$ and thus will contribute to a histogram bin corresponding to the tail of the Gaussian distribution. If both $\boldsymbol{x_i}$ and $\boldsymbol{p_i}$ belong to the overlapping region, then the distance between them is likely small and generally contributes to a histogram bin close to the peak of the Gaussian. Zhang uses a heuristic method to determine a threshold such that any point pairs that have distances larger than this threshold are considered outliers and removed from the registration calculations for that ICP iteration. This threshold is determined based on the range image sampling distance, the mean point pair distance, and the standard deviation of the distances.

In addition to ICP-based registration approaches, many researchers use other iterative whole-surface registration techniques. The most popular variants are based on Chen and Medioni's work published the same year as Besl and McKay's ICP paper [3],[6],[12], which minimizes the distance along point normals in one data set to tangent planes in the other data set. Eggert *et al.* [7] use a spring-mass model to perform registration.

In addition to addressing the issue of pair-wise registration of range images, researchers are becoming more interested in being able to register large numbers of range images to form complete models [3],[6],[7],[12]. In the remainder of this paper, we will concentrate on pair-wise registration because the most popular multi-view approaches have critical pair-wise registration components to them.

3. A Registration Test-Bed

3.1. Rationale and Goals for the Test-Bed

Many of the papers discussing variants of the ICP algorithm and other registration algorithms have only been tested on a small set of objects. Additionally, few algorithms have been tested on the same set of data. Broader comparative studies to date have been limited in scope, such as Lorusso's studies on four different methods of calculating the ICP registration values [11]. This lack of comparative data makes it difficult to judge the true strengths and weaknesses of each algorithm. A test-bed platform, if properly designed, facilitates effective comparisons between algorithm variants. The following goals serve as the basis for such a design.

The *extensibility* of the platform is measured by the ease with which it may be extended to include other registration algorithms and algorithm variants. This ease translates into decreased development time for new variants. Since there are so many algorithms, variants, and sub-variants for registration, this feature is important.

After implementing the test-bed, it must be used to gather data to evaluate the relative fitness of various algorithm variants. A platform is said to be *instrumentable* if a wide range of performance data can easily be gathered from it. At a minimum, the test-bed must be able to yield readily the following information given a set of registration methods and input data:

1. Final registration error
2. Registration transformations
3. Number of ICP iterations
4. Total execution time

The final transforms and registration error provide the means for evaluating the validity and quality of a particular registration test. The registration error serves as an objective value, and the transformations allow for visualization of the results for subjective human verification of the registration. The total number of ICP iterations required and the total execution time provide the means to evaluate the relative performance of registration tests.

Given a highly instrumentable and extensible set of software for supporting the registration of range images, the test-bed software should be engineered to be *reusable* in larger applications. For example, an application that merges range images should be able to use the registration software built for the test-bed as a pre-processing step.

3.2. Implementation

Our range image registration test-bed software uses the *Visualization Toolkit*, an open-source library for the manipulation and visualization of 2D, 3D, and higher-dimensional data [14]. The library contains an object hierarchy built to support componentized visualization pipelines. Our software builds upon the base library by supplying a pair-wise ICP registration algorithm with pluggable variants to the base algorithm. The key ICP variants currently implemented and tested include:

1. *ICP Iteration Control:* Uses Besl's criterion requiring the change in the mean squared surface between two ICP iterations to drop below a pre-specified level [2].
2. *Outlier Point Classification*
    a. *Schütz's Distance Thresholder:* Identifies outliers as those point pairs that are separated by "too much" space in an attempt to solve the problem of partially–overlapping data sets [15].
    b. *Zhang's Statistical Outlier Classifier:* Examines a histogram of unsigned point pair distances to estimate which pairs are outliers [18].

In addition to these variant classes, our test-bed contains infrastructure to support additional ICP variants and other, non-ICP registration techniques. Also, we have written additional software to aid in the visualization and in the numerical evaluation of registrations.

4. Experimental Analysis and Results

In the absence of actual "ground truth" registrations against which comparisons may be made, we used a combination of qualitative and quantitative measures to analyze the experimental results. Qualitatively, we examined a sampling of the test data to verify trends and give meaning to the numerical results. We used the numerical data to guide this sampling process and produce additional statistics on the registration results.

4.1. Experimental Setup

We selected several views from four real-world objects (Figure 1) as a test database. From this set of views, pairs from an object were selected, then we used the GUI to produce an initial registration. For the tests we performed, we attempted to produce "good", realistic initial registrations that visually appeared to only need a small amount of fine registration.

Once the initial registration was created, we saved the base test configuration file to disk. This configuration file was then duplicated and modified for each registration variant permutation desired. The variables upon which these permutations are based for our tests are data set, view pair, decimation factor, outlier classifier, and loop exit criterion threshold.

Our tests are named according to which values for these parameters were chosen. The remaining subsections will give details on the permutations chosen. The final subsection of this section summarizes the permutations and briefly describes our registration test environment.

For this paper, we acquired range images from the "Angel", "Buddha", "FacesPat1", and "FacesPat2" data sets. All of the range images on which we gathered data are shown in Figure 1. We used a Minolta Vivid 700 range sensor to acquire these range images. Most of the tests included the 000 image as one of the two images in the pair. The narrowest view angle used in these tests was 20 degrees and the widest view angle was approximately 126 degrees. The most important feature of "Angel" data set is that the main view shown in Figure 1 is, for all intents and purposes, a superset of all of the other views. The large wings on either side of the body block the camera's view of the side and back of the angel. In the "Buddha" data set, the Buddha head is quite round and has some very nice 3D texture in the hair. The ears served as one of the locations in which it was easiest to find misregistrations."FacesPat1" and "FacesRick1" proved to be smoother data sets than the Buddha images, however they provided more prominent noses and more complex ear structures. All of the range images plus many others are available in our range image database

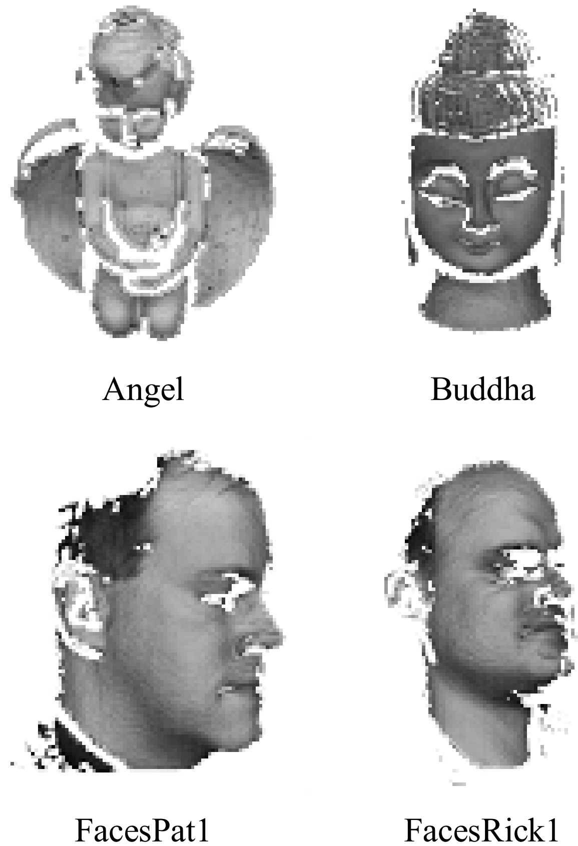

Figure 1: Selected texture-mapped rendering of the range images used in our experiments.

located in the OSU SAMPL web site at http://sampl.eng.ohio-state.edu/.

We tested our experiments at different uniform decimation factors, including 1, 2, 4, 8, 16, and 32. Larger decimation factors indicate smaller tested images (a 200x200 range image decimated by a factor of 4 is 50x50).

Three main algorithms were used for our tests to perform outlier classification of corresponding point pairs. First, the base, parameter-less algorithm introduced by [2] that classifies no points as outliers was used. Additionally, classifiers based on [15] and [18] were also used. These latter two were provided the same set of values for the expected registration error. We used a base value of 3.5492mm for each of these, plus several multiples of this value. This base value was obtained from the mean edge length of several initial data sets at a decimation factor of 2.

The loop exit criterion threshold is the maximum difference between successive registration errors required to consider an ICP sequence to have converged. We chose the loop exit criterion values of `0.3`, `0.03`, and `0.003` somewhat arbitrarily because they yielded good results with initial tests. Except when analyzing the effects of the criterion value, the tightest threshold, `0.003` was generally used for qualitative tests.

After generating the test configuration files, four 450MHz Pentium II™ machines with 256MB of RAM were used to perform batch experiment runs. We collected data from a total of 7,699 tests for analysis, which consists of four different input objects with a total of 18 view pairs. For each of these tests, we recorded the configuration, key data from each ICP iteration, and the total execution time. From the results files, we compiled a database of the final registration information, the execution time, and the number of iterations required for each test.

4.2. Analysis Methodology

After performing batch run tests of our registration experiments, we collected the results and examined key individual tests in the GUI. Most of the examinations performed in the GUI were made to characterize how the outlier classifier choices affect the quality of the registration. Typically, the no-outlier results were viewed as a baseline, then results from the two classifiers were viewed. The following criteria were generally used to judge the quality of a registration in the GUI:

1. Are there any gross registration errors?
2. Are there any mismatched edges?
3. Are there "splotchy" sections?

Gross registration errors consist of registrations that are completely wrong. For example, Figure 2 shows two face range images where the nose from one is pointing out the ear of the other. Occasionally, obvious registration errors such as these correspond to very low registration error values because, even though the registration is incorrect, the registration produces a low mean squared error.

If the entire registration is not obviously incorrect, we next looked for key feature areas such as ears and noses on the face images because they were easy to examine with polygonal rendering. Generally these feature areas would have edges in one or more range images allowing us to more easily see between the two range images. Often, we instructed the GUI to draw lines connecting the corresponding point pairs. Figure 3 shows a case where the angel's right wing is misregistered. The pink edge of the wing intersects the cyan edge instead of being aligned with it.

Finally, if there were no problems found in these feature areas, we examined large areas with relatively constant curvature. Given a perfect registration of range images that have Gaussian noise, we expected that the two surfaces would cross over each other often, creating a "splotchy" surface as in Figure 4. A worse registration would not have this characteristic because the two surfaces would be too far away to have this interleaving, as shown in Figure 5. Thus, we generally considered slightly splotchy surfaces to have a better registration than registrations displaying large expanses of non-interweaving range image sections.

Using these criteria, we qualitatively analyzed the registration results to determine which algorithm variants worked the best, and under what conditions. Additional quantitative results were analyzed. Key findings from those analyses are found in section 4.3.

4.3. Experimental Results

Upon examining our results, we determined the key effects of each of the experiment permutation variables. We have broken down the results of these tests into the following categories:

1. Effects of Outlier Classifier Type and Parameter Settings
2. Effects of Decimation
3. Effects of the Loop Criterion Threshold Value

For each of these effects sections detailed below, we will highlight key similarities and differences with the data sets we tested.

4.3.1. Effects of Outlier Classifier Type and Parameter Settings. One aspect that significantly affects the number of iterations and the total execution time of a test is whether the ICP process is deemed invalid before it converges. When no classifier is used, 84% of the tests converged before an ICP iteration resulted in a worse registration than the previous iteration. When either of the two outlier classifiers were used, this rate drops dramatically to approximately 25%. We hypothesize that the following is happening:

1. The outlier classifier marks outlier point pairs based on a registration.
2. The next registration is calculated and applied given the set of inlier pairs. This registration results in a small movement of the range image.
3. The outlier classifier marks outlier points, but this time it selects a significantly different set of pairs as outliers.
4. The next registration is calculated and applied on the new set of inlier pairs. The different set of inliers results in a registration error greater than that calculated in step #2.

Figure 2: Rendering that demonstrates a catastrophic failure of the registration when large non-overlapping regions exist and no corresponding point pairs are classified as outliers

Figure 3: Rendering that demonstrates a mis-registration at the edge of the wing in the circled region.

Figure 4: Rendering of a registration that demonstrates good "splotching."

Figure 5: Rendering of a registration that demonstrates poor "splotching."

We have found that often if we do not check that each ICP iteration's error value is less than the previous iteration's, the ICP cycle enters an infinite loop, apparently jittering between different sets of inliers and outlier alternately pulling the registration different directions upon different ICP iterations.

Regardless of whether the ICP sequence was deemed to have converged, we found the effects of using an outlier classifier to be generally as expected. When nearly the entire surface from the range image being registered overlaps the other range image, we found that not using any classifier produced the best results. We found this situation to be the case for the "Angel" test set as well as for range image pairs that were only separated by a small angle.

The "Angel" test set approximated Besl and McKay's original requirement of always registering a subset of an object to the object in the following way. The base range image to which the other range images were registered was the frontal view of the angel. This view captures the face, wings, and front of the body, only missing some of the sides of the body that form oblique angles to the camera. All other views could not see much more of the original data because the wings blocked the sides and the back.For the other data sets, as the amount of non-overlapping data increases, the classifiers become more important. For most of the tests examined, we found that the classifier based on Zhang's work [18] performed the best, though for many tests, it was only slightly better than the one based on Schütz *et al.* [15].

A major deficiency in the algorithms we tested was that they generally had difficulty registering certain human-identifiable features of high curvature changes and edges of range images. For example, the tips of the ear lobes in the face tests would generally be closely registered for the most successful tests, but often the curves at those tips did not match correctly. We expect that including the smoothed normals as [7] suggests would assist in obtaining better results.

4.3.2. Effects of Decimation. In general, we found that lower decimation factors produced better results, though they required greater execution time. In particular, the simplistic brute-force nearest-neighbor search used for finding closest point pairs proved particularly slow for undecimated data sets due to its $O(N^2)$ complexity. For example, one of the Buddha tests took approximately 1 minute, 45 seconds to load into the GUI. The bulk of this time was spent performing the nearest neighbor search. After breaking up the input points into uniformly sized bins to improve search speed, we saw this loading time decrease to 40 seconds. In order to not skew the timing results for later tests, the original $O(N^2)$ algorithm was used for all tests and the enhanced search was only used for the GUI visualization. We believe that a more sophisticated search such as a kd-tree based search [8] would yield much faster results.

As for the registration quality, when each test is viewed with the decimation factor used to perform the registration, the results tend to be quite good. Unfortunately, the uniform decimation is sub-optimal for preserving the original shape. If a registration performed on decimated data is viewed with undecimated data, the more subtle features of the shape show up and expose mis-registration problems. We found that because the initial course hand registrations were close enough to the correct registration that decimation factors above 2 produced unsatisfactory results. Higher decimation factors generally resulted in fine registrations that were qualitatively worse than the original course registration, when viewing the results with undecimated data.

We did not find any strong correlation between the decimation factor and how the outlier classifier parameters affected whether an ICP iteration terminated successfully or became invalid

4.3.3. Effects of the Loop Criterion Threshold Value. Modifying the loop criterion produced predictable results. As tighter criteria were used, we noticed greater refinement in the registration path being followed. If the ICP algorithm was moving toward an incorrect registration, a tighter criterion simply allowed it to move in closer to that incorrect registration. Additionally, the tighter the criterion, the greater chances there were that the ICP sequence would become invalidated.

5. Summary and Conclusions

In this section, we have described our test setup and the generation of the 7,699 tests for our experiments. We have proposed a method of analyzing the quality of range image registrations, and given our analysis of our experiments using this method.

In that analysis, we found that for range image pairs that approximate Besl and McKay's requirement of full overlap, using no outlier classifier generally yielded the best results. For those pairs that had significant non-overlapping regions, both of the classifiers generally yielded good results, with the classifier based on Zhang's work performing slightly better in most cases than the one based on the work of Schütz.

We also found that although decimated data could be registered, those registrations tend to only be "good" in the context of their decimation. Once the range image pair is viewed undecimated, the registration imperfections readily manifest themselves. On the flip side, due to our rather brute-force approach to finding closest point pairs, decimation had an extremely significant impact on the execution time required. Additionally, we found that modifying the threshold for determining convergence of an ICP sequence had predictable results. As that threshold is lowered, the sequence simply goes further along the path it is following unless it first encounters numerical or algorithmic instabilities.

Through performing the tests and analyses, we benefited from the design goals that we followed. The hooks for instrumentation allowed us to gather the required data from our experiments. Further, we were able to add new variants as necessary to our test-bed due to its extensible nature. Internally, we reused code as we developed classes such as one that creates lines to visually connect closest point pairs. That class shares the point correspondence filters with the master ICP registration class, and as we built multiple applications that shared the library classes discussed in this paper.

6. References

[1] P.J. Besl . "Active Optical Range Imaging Sensors". *Machine Vision and Applications*, 1(2): 127-152, 1988.

[2] P.J. Besl and N.D. McKay. "A Method for Registration of 3-D Shapes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239-256, Feb. 1992.

[3] Y. Chen and G.G. Medioni. "Object Modeling by Registration of Multiple Rnage Images". *Image and Vision Computing*, 10(3):145-155, 1992.

[4] C.S. Chua and R. Jarvis. "3D Free-Form Surface Registration and Object Recognition". *International Journal of Computer Vision*, 17(1):77-99, January, 1996.

[5] B. Curless and M. Levoy. "A Volumetric Method for Building Complex Models from Range Images". *Proceedings of SIGGRAPH*, 303–312, August 1996.

[6] C. Dorai, G. Wang, A.K. Jain, and Carolyn Mercer. "Registration and Integration of Multiple Object Views for 3D Model Construction". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):83–89, January 1998.

[7] D.W. Eggert, A.W. Fitzgibbon, and R.B. Fisher. "Simultaneous Registration of Multiple Range Views for use in Reverse Engineering of CAD Models". *Computer Vision and Image Understanding*, 69(3):253-272, March 1998.

[8] J.H. Friedman, J.L. Bentley, and R.A. Finkel. "An Algorithm for Finding Best Matches in Logarithmic Expected Time". *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.

[9] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. "Piecewise Smooth Surface Reconstruction". *Proceedings of SIGGRAPH*, 295–302, July 1994.

[10] V. Krishnamurthy and M. Levoy. "Fitting Smooth Surfaces to Dense Polygon Meshes". *Proceedings of SIGGRAPH*, 1996.

[11] A. Lorusso, D.W. Eggert, and R.B. Fisher. "Estimating 3-D Rigid-Body Transformations: A Comparison of Four Major Algorithms". *Machine Vision and Applications*, 9(5-6):272-290, 1997.

[12] K. Pulli. "Multiview Registration for Large Data Sets". *Second International Conference on 3-D Digital Imaging and Modeling*, 1999.

[13] K. Pulli. "Robust Meshes from Multiple Range Maps". *First International Conference on 3-D Digital Imaging and Modeling*, 1997.

[14] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit*. Prentice Hall PTR, 2nd Edition, 1998.

[15] C. Schütz, T. Jost, and H. Hügli. "Semi-Automatic 3D Object Digitizing System Using Range Images". *Proceedings of Asian Conference on Computer Vision*, Jan. 1998.

[16] I. Stamos and P. Allen. "3D Model Construction Using Range and Image Data". *Proceedings of Computer Vision and Pattern Recognition 2000*. I:531-536. 13-15 June 2000.

[17] G. Turk and M. Levoy. "Zippered Polygon Meshes from Range Images". *Proceedings of SIGGRAPH '94* (Orlando, Florida, July 24-29, 1994). In *Computer Graphics Proceedings, Annual Conference Series*, 1994. *ACM SIGGRAPH*, pp. 311-318.

[18] Z.Y.Zhang. "Iterative Point Matching for Registration of Free-Form Curves and Surfaces". *International Journal of Computer Vision*, 13(2):119-152, Oct. 1994.