Registration of Range Data Using a Hybrid Simulated Annealing and Iterative Closest Point Algorithm[1]

Authors

Jason P. Luck
Engineering Division, Colorado School of Mines
Golden, CO 80401
303-273-3666, jluck@mines.edu

William A. Hoff (corresponding author)
Engineering Division, Colorado School of Mines
Golden, CO 80401
303-273-3761, whoff@mines.edu

Robert G. Underwood
Mathematical and Computer Sciences, Colorado School of Mines
Golden, CO 80401
303-273-3977, runderwo@mines.edu

Charles Q. Little
M/S 1003, Sandia National Laboratory
Albuquerque, NM 87185-1004
(505) 284-3151, cqlittl@sandia.gov

Abstract

This paper presents a novel approach for registering two sets of 3-D range data points, using an optimization algorithm that is both robust and efficient. The algorithm combines the speed of an iterative closest point algorithm with the robustness of a simulated annealing algorithm. Additionally, a robust error function is incorporated to deal with outliers.

Index terms: Registration, range data, simulated annealing, iterative closest point, optimization, model construction.

[1] This work was supported by a grant from Sandia National Laboratory.

1 Introduction

We present a novel approach for registering two sets of range data, using an optimization algorithm that is both robust and efficient. The specific problem we are addressing is this: A sensor samples three-dimensional (3-D) points from the surfaces of objects in a scene, from two different viewpoints. The point data are represented in the coordinate system of the sensor at each viewpoint. The goal is to estimate the six degree-of-freedom (DOF) transformation between the two sensor viewpoints using only the range data, given an initial guess. Once the transformation is known, the two sets of range data can be *registered*, or brought into alignment.

Figure 1 shows an example of range data, taken by a structured light sensor at Sandia National Laboratories. The data illustrates some problems that may be encountered when performing registration. First, the two scans are only partially overlapping; *i.e.*, we do not have the case where one scan is only a subset of the other. The data is noisy, with non-uniform spacing, and contains outlier data points (erroneous points that are not on actual surfaces). Points that are in the non-overlapping area are also considered to be outliers. Finally, due to the discrete nature of the sampling, one cannot rely on a point to be sampled in the exact same position in each scan. These characteristics make registration a difficult problem.

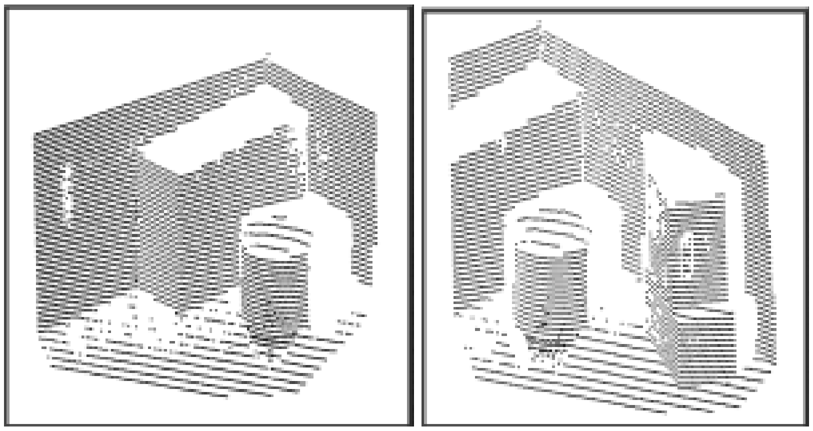

Figure 1 Two partially overlapping range data scans taken with a structured lighting system at Sandia National Laboratories with ~5000 points per scan of a 4x4 ft. corner.

Nonetheless, there are many useful applications where registration of range data is an important step. One important application is model construction, where a composite geometric model of an object or scene is constructed from a set of range scans taken from multiple viewpoints. Here, in addition to performing the registration step, we must also deal with issues such as representation of the model (e.g., surface or volumetric), and merging of the data after it has been registered. An example is shown in Figure 2. The left two images show surface models constructed from the scans shown in Figure 1. Registration is performed to bring the two models into alignment. These are then merged together to form a final model, as shown in the rightmost image. Since it is relatively easy to construct a surface model from point data and to merge the surface models together once they have been registered, we will focus on the registration process in this paper.

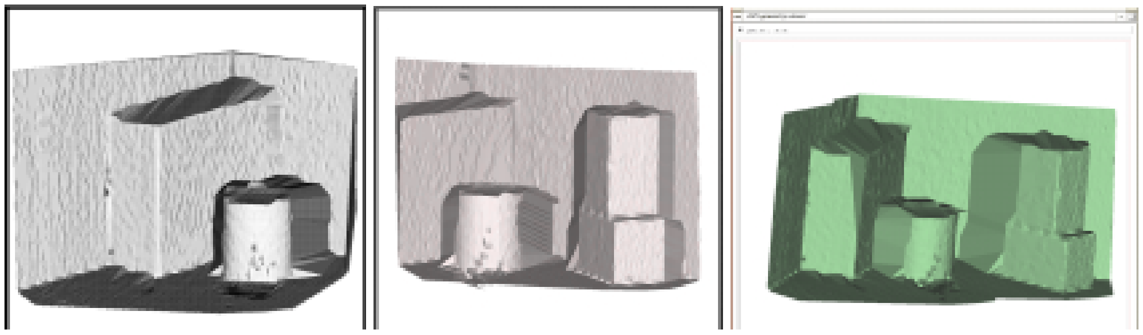

Figure 2 The left two images show surface models created from the point data in Figure 1. The rightmost images shows the final fused surface model created after registration.

In some applications, one can precisely calibrate the extrinsic parameters of the sensors (*i.e.*, their position and orientation in the world), so that it is not necessary to estimate the registration parameters from the range data. However, in many applications, these are not known well enough for the desired task, although a course estimate is often available. For example, a robot moving around in a room may have an estimate of its position and orientation from dead reckoning (odometry), but this is usually not accurate enough to create an accurate model of the room. Therefore, it is necessary to use the data itself to refine the registration estimate.

Due to the importance of registration, much work has been done in the past on this topic (a recent review is given by Goshtasby [1]). Some approaches rely on identifying a set of distinctive features in the two range images (such as curvature extrema [2] or surface patches [3]), and using the correspondences between them to compute the transformation. However, the accuracy of the registration depends on the accuracy of the extracted features and whether they can be reliably extracted from both images.

Alternatively, one can match the entire data set from one scan with that from the other. These approaches commonly use an optimization procedure that starts with an initial guess of the registration parameters, and iteratively refines the registration by reducing the error between corresponding points in the two scans [4] [5] [6]. Since a point in one scan may not have an exact match in the other scan, it is necessary to match it to the closest point on a local surface approximation in the other scan.

The iterative closest point algorithm (ICP) [5] is a commonly used registration algorithm of this type. The process is designed to register two sets of data points, where one set is a rotated and translated subset of the other. The algorithm uses a closest point estimation to compute correspondences. That is, for every data point from one set, the point from the other data set that is geometrically closest is taken as its corresponding point. (An alternative is to take the closest point on a surface approximation to the model points [4].) The process then repetitively uses the absolute orientation algorithm [7] to register the point sets; *i.e.*, to transform one data set so that it is aligned with the other. The absolute orientation algorithm finds the transformation that yields the least squared error distance between corresponding points. The algorithm steps are as follows:

1. Determine correspondences by computing, for each point of one data set, the closest point to the other data set.
2. Compute the transformation using the absolute orientation algorithm.
3. Apply the transformation to register the sets.
4. Repeat steps 1–3 until the change in the error falls below a threshold.

This process slowly pulls the data points in to the model points, and monotonically decreases the error. However, ICP will only converge to the closest local minimum of the error surface. Thus, the algorithm may never find the global minimum, thus missing the correct registration.

In our domain, the error function usually has a single global minimum surrounded by many local minima. Figure 3 shows a plot of the error function (to be described in the next section) as the registration parameters were varied over 2 of the 6 dimensions (one translational and one rotational dimension), using the “corner” data set from Figure 1). The resulting plot clearly shows a number of local minima contained in a larger pit. Of course, this is only a 2-D projection of the true 6-D surface, and there could be many more local minima in 6-D.

Accordingly a registration algorithm must be able to deal with local minima, such as through the use of a stochastic optimization algorithm like simulated annealing (SA) [8]. Simulated annealing has been used by a number of researchers for registration and model fitting [9] [10]. The technique is able to escape local minima, but is extremely slow in converging to the global minimum.

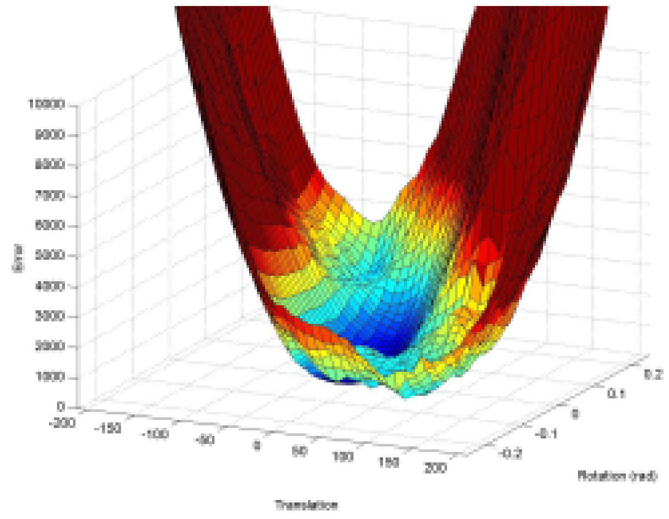

Figure 3 The error function was plotted as two registration parameters were varied.

Thus, when dealing with an error surface such as in Figure 3, ICP is efficient but will get stuck in local minima, while SA is robust but inefficient. This suggests the development of a hybrid technique in which the two methods are combined so that local minima can be crossed while moving to find the global minimum as efficiently as possible. One way to do this is to use SA to provide “good” starting points for ICP. Rather than exhaustively trying many uniformly spaced or randomly placed starting points, SA can more intelligently provide starting points closer to the global minimum. In this way, SA provides “guided” restarts that get ICP out of local minima and closer to the actual solution. Then ICP is used to move efficiently down the error surface.

We developed a hybrid registration algorithm, combining both an efficient local minimizer and a robust global search algorithm, to test these concepts. The algorithm was tested on a series of synthetic and real range data sets, and was found to escape local minima while remaining efficient. The algorithm is described below in Section 2, and the results are described in Section 3. Finally, Section 4 offers some concluding remarks.

2 Description of System

Although the focus of this paper is on the optimization problem, we developed a complete registration and modeling system. An overall description of the system is as follows (details are covered in [11]): First, data acquisition is performed to acquire the range scans. A surface model of each scan is created, using a Delaunay triangulation (although other surface models could be used). Using the coarse (dead reckoning) position estimate, the overlapping regions of the two sets are estimated and segmented out, using a novel frustum segmentation technique. Next, registration is performed on the segmented regions using the hybrid algorithm, which incorporates a robust error function that can deal with up to 50% outliers (outliers are erroneous points that are not on actual surfaces, or points that are in the non-overlapping area). Finally, the two surface models are stitched together to yield the final model.

2.1 Finding the Overlapping Regions

The first step in the registration process is to estimate the overlap between scans, because only this overlap will be used for registration. The process assumes that an initial guess for the registration is available, such as through dead reckoning. This initial guess is used to execute an initial registration of the two scans. We can then identify regions in the two scans that overlap using the novel frustum technique described in the next paragraph. Of course, due to errors in the initial registration, the overlapping regions will not overlap perfectly. However, we assume that the initial registration is good enough so that we have at least 50% actual overlap. We can then use the points within these regions to perform the registration.

The frustum segmentation technique we developed to segment the overlapping regions is briefly described here (details are covered in [11]). To determine the overlapping region for one of the scans, we create a frustum (pyramid) using the position of the scanner and the outline of the points visible from the scanner in that position. The frustum is simply a set of triangles extending from the scanner position around the outline of the point set. Since the Delaunay triangulation creates a convex hull around the point set, the frustum is also convex. Therefore, any point within the frustum will be on the inside of each frustum plane. This frustum is then projected onto the points from the second scan, and only points within the frustum are retained as part of the overlapping set (Figure 4). The process is then switched to find the overlapping region for the other data set. We are currently testing a new technique that allows for non-convex surfaces.

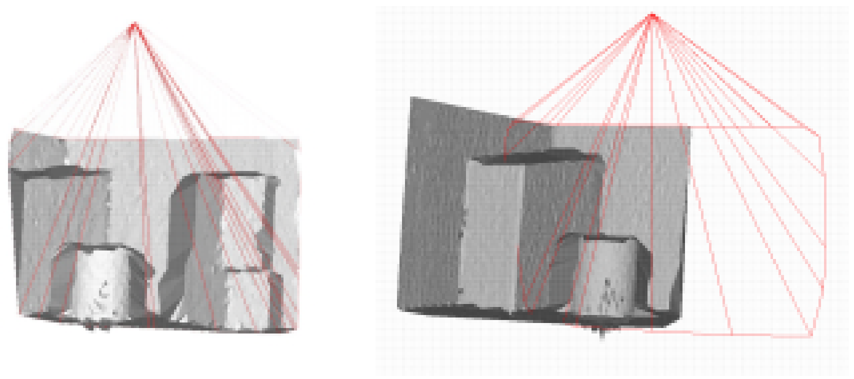

Figure 4 A frustum (pyramid) is drawn from the sensor location to the outline of the observed data points (left figure). The frustum is projected onto the other scan using the initial registration guess (right figure).

The result of this segmentation process is two sets of points, of which we assume that at least 50% are in the common area between the two sets. The purpose of this step is to cull out portions of the range data that we know do not overlap, and therefore reduce the number of "outlier" data points that we have to deal with in subsequent processing. This allows for accurate registration by minimizing a robust error function, described in the next section.

2.2 Robust Error Function

The error function, $E(\mathbf{T})$, should be at its minimum when we have the correct registration parameters for the transformation $\mathbf{T}$ between the scans. If $\mathbf{p_i}$ is a point from one scan, then $\mathbf{T}(\mathbf{p_i})$

is the same point expressed in the coordinate frame of the other scan. If correctly registered, then $\mathbf{p_i}$ should ideally lie very close to a corresponding point $\mathbf{q_i}$ from the other scan. However, there is no guarantee that the scanner will sample a data point $\mathbf{q_i}$ in the exact same place on the surface as the original point $\mathbf{p_i}$ in the first scan. Therefore, we must match points from one scan to a surface model of the other scan [4] [12]. We compute the closest point $\mathbf{c_i} = C(\mathbf{p_i}, \mathbf{Q})$ to the triangulated surface model of the other data set, and use that as the corresponding point. The error function is based on the squared distances between corresponding points from the two sets, $d_i^2 = \left\|\mathbf{c_i} - \mathbf{T}(\mathbf{p_i})\right\|^2$.

However, simply taking the mean squared error for all points will result in an error function that is susceptible to the effect of outliers. In other words, incorrectly matched points that have a large error distance will have a large effect on the total score. To avoid this, a robust estimation scheme is needed [13].

There are many approaches to robust estimation. One approach is to use the least median of squares (LMS) estimator [14]. In this way, up to 50% of the points can be outliers. Another approach is to weight each point, so that points with large errors contribute a small amount to the total score. The error function is then calculated as:

$$E(\mathbf{T}) = \frac{1}{N}\sum_{i=1}^{N} w_i d_i^2 \tag{1}$$

For example, Zhang [15] dynamically computes a threshold based on the distribution of the distances, such that points beyond the threshold are assigned a weight of zero. We use a similar approach, using weights w_i that are dependent on the median of the squared distances from each data point to the surface model. However, we do not let weights go to zero for outliers. We define a threshold $t = 2 * median\left(d_i^2\right)$, and assign weights as follows:

$$w_i = \begin{cases} 1 & if\ d_i^2 < t \\ t / d_i^2 & if\ d_i^2 \geq t \end{cases} \tag{2}$$

Accordingly good points have a weight of 1, but outliers are weighted such that they contribute a constant amount to the error (Figure 5). In this way, we can have up to 50% outliers without affecting the value of the threshold *t*. We have found that the resulting error function can achieve excellent registration even in the presence of outliers of up to 50% of the data.

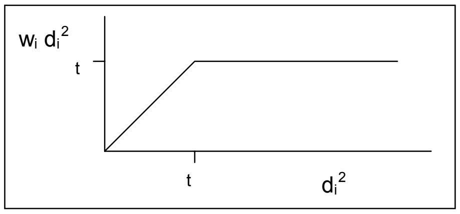

Figure 5 The error contribution from a single point, $w_i\ d_i^2$, as a function of the squared distance, d_i^2.

The error function assures that errors contributed by outliers do not grow without bound, but are limited to a small but positive amount. The reason we do not give a zero weight to outliers is that if we did, the algorithm may tag entire surfaces as outliers. For instance, consider the case of a scene consisting of three orthogonal planes, such as the corner of a room. If the error is allowed to go to zero, the lowest error score for the corner of a room is achieved when two planes are aligned and the third is hovering above the threshold. This occurs because in this position the hovering points will have an error of zero, but if aligned correctly the points would contribute a small amount of error due to noise in the data.

2.3 Optimization Algorithm

As previously stated, the novel hybrid optimization algorithm is a combination of the ICP and SA algorithms. The form of SA used is a variation of the Nelder-Mead downhill simplex method, which incorporates a random variable to overcome local minima [16]. A simplex is simply a set of *N*+1 guesses, or vertices, of the *N*-dimensional state-vector sought and the error associated with each guess. In our case we have *N*=6 corresponding to the six degrees of freedom of the registration parameters. The simplex attempts to walk downhill by replacing the vertex associated with the highest error by a better point. The algorithm uses a random element, based on a "temperature" parameter, to escape local minima. This is done by subtracting a random amount, scaled by the temperature, from each tested replacement point. Therefore while any move that is a true downhill step will be accepted, some additional uphill steps will also be accepted. At high temperatures most moves are accepted and the simplex roams freely over the search space. At lower temperature only smaller uphill steps can be accepted. As the temperature is slowly lowered the simplex crawls out of local minima and converges upon the global minimum.

The ICP algorithm used is the standard algorithm described earlier, with the exception of one modification. We need to have both ICP and SA operate on the same error surface. To do this the algorithms must use the same error function. Recall that the robust error function incorporated into the SA algorithm weights each distance so that outlier points have a lesser effect. ICP repetitively calls the absolute orientation algorithm to calculate the transformation parameters. Accordingly the absolute orientation algorithm had to be modified to include weighting of the point pairs. The weights are determined by the robust error function, and are incorporated into the absolute orientation algorithm as described by Horn [7]. The result is that ICP operates on the same error surface as SA.

Our hybrid algorithm alternates between the SA and ICP algorithms described above. Effectively, SA is used to find good starting points for ICP. In other words, ICP does the majority of the work, but when trapped in a local minima SA will attempt to traverse the minima and choose a new starting point for ICP. A flow chart is shown in Figure 6.

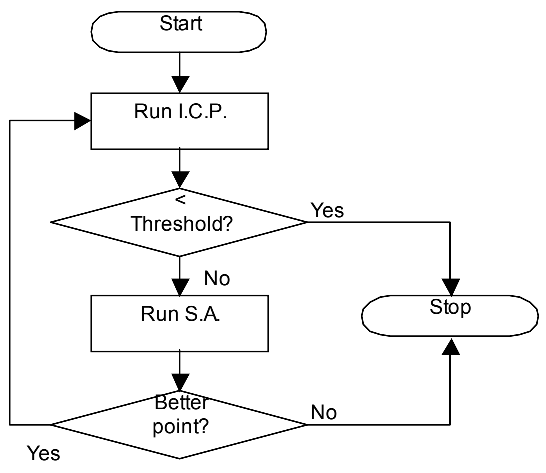

Figure 6 Flow chart for the hybrid algorithm.

The process starts by running ICP from the original position, which will converge to the nearest local minima. If the error score is below a pre-set error threshold (discussed below), then the algorithm stops. Otherwise, SA is used to search about the error surface for a point with a lower error score. The starting point for SA is initialized to the point at which ICP stopped. SA will continue until a better point is found, where it relinquishes control to ICP, or until the temperature falls below a minimum temperature threshold, indicating that SA has run long enough and was unable to find a better point.

One issue is how to set the error threshold in the algorithm above. By setting the threshold, we assume that we can recognize when ICP has found the global minimum and we should therefore stop searching. This speeds up the process by preventing additional searching by SA for better starting points. The threshold can be set by estimating the lowest expected error that one would have for two scans in perfect registration. This is dependent on the accuracy of the scanner. Of course, one can always use zero for the threshold in which case the algorithm will still find the global minimum, but waste some additional time searching for a better solution.

Figure 7 depicts how this process moves across the error surface. The left figure shows a contour plot of the error surface shown in Figure 3. The hybrid algorithm was run on the "corner" data set (Figure 1), and the route of the algorithm was projected onto this error surface.

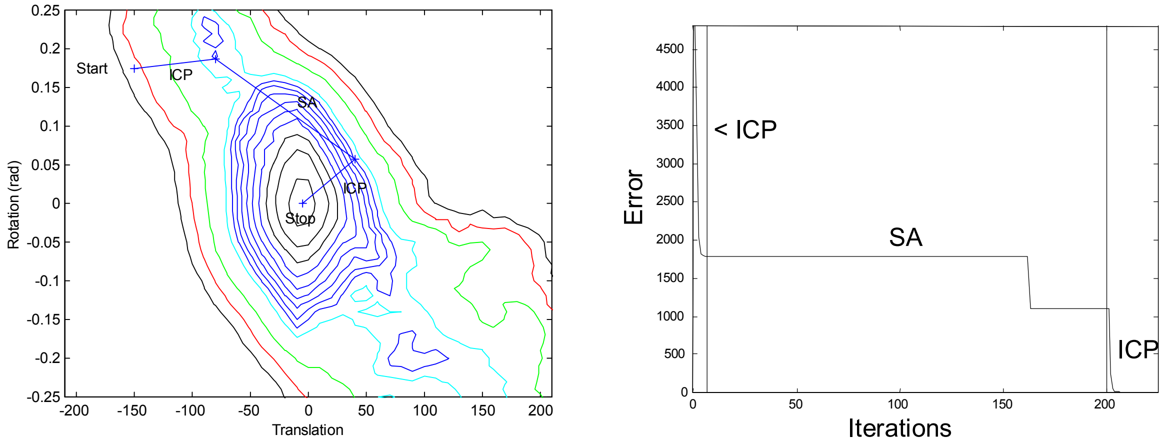

Figure 7 Left: Solution path of the hybrid algorithm on a 2-D projection of the error surface. Right: error score as a function of the number of iterations as

The figure shows the effectiveness of the hybrid algorithm. As can be seen ICP initially converges on a local minimum. Then SA moves to a better point, from which ICP is able to swiftly converge to the global minimum. The right figure shows a plot of the error score as a function of the number of iterations. As one can see, ICP takes very few iterations to initially decrease the error, but gets stuck in the local minimum. SA then finds a lower point after many iterations, and subsequently ICP quickly finds the global minimum.

3 Results

The algorithm was tested using synthetic data sets and three real data sets. Two of the real data sets were taken by a structured light range sensor at Sandia National Labs (Figure 1 and Figure 8). The other was taken by a Coleman laser range finder (Figure 9).

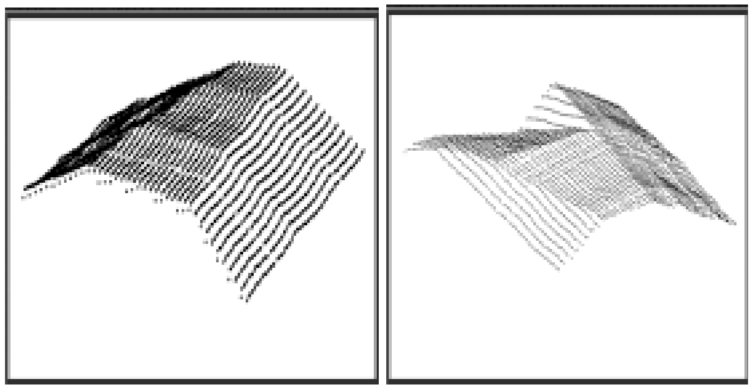

Figure 8 Structured light range data of a 3x3 ft portion of a plane wing with ~3000 points per scan.

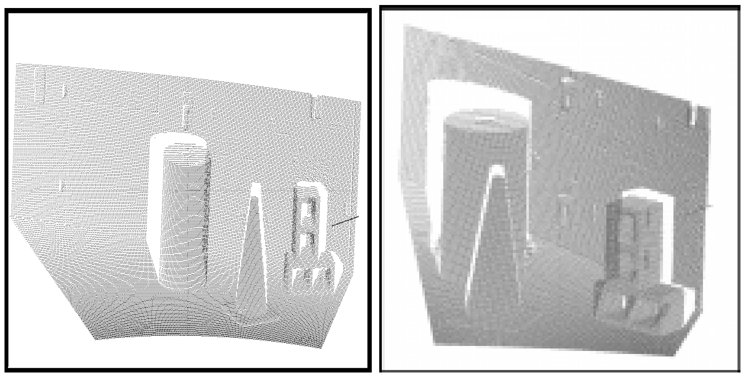

Figure 9 Laser range data of a 6x4 ft scene with ~60,000 points per scan.

The real data sets are summarized in Table 1. Note that we subsampled the data in some cases in order to speed up the algorithm. As a rough indication of running time, a complete run of the hybrid algorithm required about 10 minutes on an SGI R4400 Indigo 2.

Table 1 Summary of the three range data sets used.

	Corner (Figure 1)	Wing (Figure 8)	Coleman (Figure 9)
Points, scan 1	5012	2748	60635 (reduced to 6042)
Points, scan 2	5935 (reduced to 321)	3847 (reduced to 341)	68967 (reduced to 213)
Overlap points	165	80	110
Error function at correct registration	1.20 mm^2	1.20 mm^2	6.02 mm^2

First, the hybrid algorithm was compared to ICP by itself and SA by itself. Each method was employed ten separate times from three initial positions using the “corner” data set (Figure 1) and the “Coleman” data set (Figure 9). The results are shown in Table 2.

Table 2 Comparison of hybrid, ICP, and SA algorithms averaged from 3 starting poses spread across 5 degrees of rotation and 250 mm of translation.

Data Set	Algorithm	Average error (mm^2)	Average # iterations
Corner	Hybrid	1.31	1479
	S.A.	1.35	6508
	I.C.P.	2.21	31
Coleman	Hybrid	6.61	2301
	S.A.	8.34	6419
	I.C.P.	14.8	37

The results indicate that the hybrid algorithm achieves the approximately the same level of accuracy as the SA algorithm, while both algorithms are significantly better than the ICP algorithm. The higher error of the ICP algorithm is due to its falling into local minima. However, the results also show that the hybrid algorithm takes on the average only 23% of the iterations required by SA. Thus, the hybrid algorithm is able to achieve the same level of accuracy as SA, and does so in about 1/4th the time.

Further tests of the hybrid algorithm were performed on the “corner” and “Coleman” data sets. For each data set, the algorithm was run from five starting positions with initial errors of 5° of rotation and 50 mm of translation. One of the initial starting poses is shown in Figure 10.

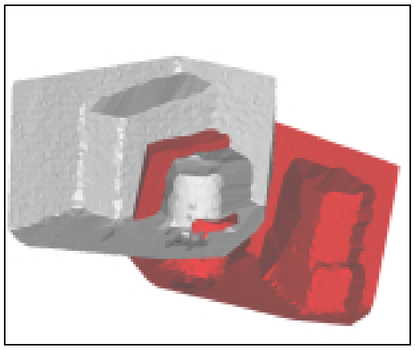

Figure 10 An initial starting position with a significant “dead reckoning” error.

Results from the two data sets are summarized in Table 3 and Table 4, respectively (additional details can be found in [11]). The tables show, for each run, the final error score that the algorithm achieved, the number of iterations required by the ICP and the SA algorithms, and the number of times that ICP fell into a local minima.

Table 3 Results from the "corner" data set.

Run	Final Error (mm^2)	Iterations (ICP + SA)	Local Minima
1.	1.38	28 + 1816	5
2.	1.28	40 + 1714	5
3.	1.29	28 + 1614	8
4.	1.37	27 + 1417	4
5.	1.21	35 + 1210	5

Table 4 Results from the "Coleman" data set.

Run	Final Error (mm^2)	Iterations (ICP + SA)	Local Minima
1.	6.25	28 + 1816	3
2.	6.03	40 + 1714	2
3.	6.46	28 + 1614	4
4.	16.07	27 + 1417	6
5.	6.08	35 + 1210	6

An interesting point is that although the final poses are extremely close, the algorithm failed to converge to the same exact location when run from different initial positions. This reveals that our error surface has many small local minima surrounding the global minimum. However, our threshold was set such that we recognized these local minima as the global minima. This is discussed in more detail in the conclusions.

When the algorithm was run on the "wing" data set (Figure 8), poor results were achieved. Since the overlap is ambiguous in at least one direction and many local minima occur along this direction, the algorithm had trouble traveling along this direction to find the global minimum. Accordingly the algorithm can be expected to perform poorly when an ambiguous overlap is used. However, any optimization algorithm will have difficulty in these cases.

An obvious alternative to the hybrid algorithm is to run ICP from many different starting points (either uniformly spaced or randomly chosen), as originally suggested by Besl and McKay [5]. However, if there are many local minima, the chance of starting the algorithm close to the global minimum is very small. To illustrate this, we created a synthetic range data set, in the form of a "bowl" shaped object with many small dimples (Figure 11). We sampled points from different locations on the surface to create two separate data sets, and added random noise to the data points. We then attempted to register one instance of the object with another.

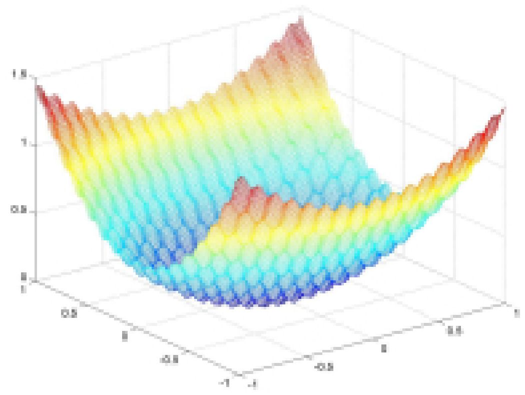

Figure 11 A synthetic "bowl" shaped object with many small dimples.

The ICP algorithm was run on this data set and two real data sets, using random restarts in an effort to find the global minimum. For comparison, SA and the hybrid algorithm were also run on the same data set from the same starting poses. As expected, the hybrid algorithm took the least number of iterations while ICP with random restarts took a very large number of iterations before the global minimum was found (Table 5).

Table 5 Performance of the algorithms on the synthetic "bowl" data and two real data sets.

	synthetic "bowl" data		real "corner" data		real "Coleman" data	
Algorithm	Error (mm^2)	Iterations	Error (mm^2)	Iterations	Error (mm^2)	Iterations
Hybrid	1.97	1815	1.32	2404	6.38	3023
S.A.	1.97	6249	1.35	6508	8.34	6419
ICP with random restarts	1.97	30401	5.14	11733	8.20	5462

4 Discussion

The main contribution of this paper is a novel optimization algorithm for registering two sets of range data. A hybrid algorithm was developed, utilizing an ICP algorithm and an SA algorithm working together. The resulting hybrid algorithm is robust in finding the correct registration and efficient in terms of the number of iterations. The system uses a robust error function to handle outlier points. Although incorporating a robust error function into ICP and SA separately has been done before, our contribution was to incorporate the same robust error function into both routines, so that they could work together on the same error surface. Through evaluation on real and synthetic range data, we found that our system has the robustness to deal with local minima while retaining efficiency. The results show our algorithm to be much more efficient than a stochastic technique (SA) by itself, and much more accurate than a local technique (ICP) by itself.

We developed a complete surface registration and modeling system that incorporated the hybrid optimization algorithm. As part of this system, we developed a novel segmentation approach that projects a frustum onto the data set in order to identify potentially overlapping regions. We also developed a method to create a combined surface model from the data points, once registration was accomplished. This work is discussed in [11].

The method does have limitations. First, the estimate of the range sensor position must be accurate enough to ensure that there is less than 50% disparity in the segmented overlap.

Additionally the algorithm will have problems if the error surface is relatively flat with many local minima (as in the "wing" data set). This type of error surface is extremely difficult to solve, and there is no guarantee that our method will do so.

There is an existing algorithm that would speed up our process. The current method for finding the closest point on the model surface for each data point is very inefficient, and accounts for the great majority of the running time. Simon uses k-d trees and closest point caching to achieve speed improvements down to approximately 7% of the original search time [17]. The introduction of these improvements would improve the speed of the registration phase.

As mentioned in Section 3, the algorithm did not consistently converge to the same minimum for repeated trials, although the final error score was very close each time. Accordingly, small local minima must surround the global minimum. This is not necessarily a problem, since each of these solutions is equally good according to the error function. However, we might find a unique solution by re-running ICP with no weights once the algorithm has converged. Since the scans are already very close to the proper alignment outliers will be easy to detect and discard. Eliminating the weights may remove the shallow local minima from the error surface, thereby allowing the algorithm to converge to the exact minimum.

Another possible direction for future improvement would be to try to speed up convergence by initially smoothing the error function surface. For example, local averaging could be done either on the original range points, or in the error function space. It is possible that many small local minima would disappear, while the relatively few deeper minima would remain. This would allow the algorithm to find a solution more quickly. Once the algorithm had converged, optimization could be performed again with no smoothing, thus allowing the maximum accuracy in the final results. Effectively, this would amount to a "coarse-to-fine" search technique.

In conclusion, this method of registering range images proves to be extremely robust while remaining efficient. By incorporating a robust error function based around the median, registration is accurate even when presented with up to 50% outliers. Lastly the combination of a local minimizer (ICP) with a stochastic search (SA) proved to be very effective at negotiating downward sloping error surfaces with many local minima.

5 References

[1] A. A. Goshtasby, "Three-dimensional model construction from multiview range images: survey with new results," *Pattern Recognition*, Vol. 31, No. 11, pp. 1705-1714, 1998.

[2] J. Thirion, "Extremal point: Definition and application to 3-D image registration," *Proc. of Computer Vision and Pattern Recognition*, IEEE, Seattle, WA, 21-23 June, pp. 587-592, 1994.

[3] B. Parvin and G. Medioni, "B-rep from unregistered multiple range images," *Proc. of International Conference on Robotics and Automation*, IEEE, Nice France, pp. 1602-1607, 1992.

[4] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images," *Image and Vision Computing*, Vol. 10, No. 3, pp. 145-155, 1992.

[5] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256, 1992.

[6] C. Dorai, J. Weng, and A. Jain, "Optimal registration of object views using range data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 10, pp. 1131-1138, 1997.

[7] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Optical Soc. of America*, Vol. 4, No. 4, pp. 629-642, 1987.

[8] S. Kirkpatrick, J. C.D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, pp. 671-680, 1983.

[9] G. Blais and M. D. Levine, "Registering Multiview Range Data to Create 3D Computer Objects," *IEEE Transactions on PAMI*, Vol. 17, No. 8, pp. 820-824, 1995.

[10] W. A. Hoff, F. W. Hood, and R. H. King, "An Interactive System for Creating Object Models from Range Data Based on Simulated Annealing," *Proc. of Int'l Conference on Robotics and Automation*, IEEE, Albuquerque, NM, April 21-27, pp. 2559-2564, 1997.

[11] J. Luck, "Registration of Range Images Through the Use of a Hybrid Simulated Annealing and Iterative Closest Point Algorithm," M.S., Engineering Division, Colorado School of Mines, Golden, CO, 1999.

[12] D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher, "Simultaneous registration of multiple range views for use in reverse engineering," *Computer Vision and Image Understanding*, Vol. 69, No. 3, pp. 253-272, 1996.

[13] P. J. Huber, *Robust Statistics*, New York, John Wiley & Sons, 1981.

[14] T. Masuda and N. Yokoya, "A Robust Method for Registration and Segmentation of Multiple Range Images," *Computer Vision and Image Understanding*, Vol. 61, No. 3, pp. 295-307, 1995.

[15] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves and Surfaces," *International Journal of Computer Vision*, Vol. 13, No. 2, pp. 119-152, 1994.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed., Cambridge University Press, 1992.

[17] D. A. Simon, M. Hebert, and T. Kanade, "Real-time 3-D Pose Estimation Using a High-Speed Range Sensor," *Proc. of International Conference on Robotics and Automation*, IEEE, 1994.