ACCV2002: The 5th Asian Conference on Computer Vision, 23–25 January 2002, Melbourne, Australia

Robust Simultaneous Registration of Multiple Range Images

Ko Nishino and Katsushi Ikeuchi
Institute of Industrial Science, The University of Tokyo
kon@computer.org, ki@iis.u-tokyo.ac.jp

Abstract

The registration problem of multiple range images is fundamental for many applications that rely on precise geometric models. We propose a robust registration method that can align multiple range images comprised of a large number of data points. The proposed method minimizes an error function that is constructed to be global against all range images, providing the ability to diffusively distribute errors instead of accumulating them. The minimization strategy is designed to be efficient and robust against outliers by using conjugate gradient search utilizing M-estimator. Also, for "better" point correspondence search, the laser reflectance strength is used as an additional attribute of each 3D data point. For robustness against data noise, the framework is designed not to use secondary information, i.e. surface normals, in its error metric. We describe the details of the proposed method, and present experimental results applying the proposed method to real data.

1 Introduction

Registration of multiple point cloud range images is an important and fundamental research topic in both computer vision and computer graphics. Many applications and algorithms can be (are) developed on the assumption that accurate geometric models are obtained a priori, e.g., recognition, localization, tracking, appearance analysis, texture-mapping, metamorphism, and virtual/mixed reality systems in general, among others. Additionally, projects to construct precise geometric models based on observation of real world objects for the purpose of digital preservation of cultural heritage objects have drawn attention recently [3, 9, 21]. Because of their objective, these projects require very precise registration of multiple range images.

In this paper, we propose a framework to register multiple range images robustly. Taking the point cloud images obtained through use of a range sensor, e.g., laser range scanner [14, 13, 15], light-stripe range finder [24], etc., as the input, we simultaneously register all range images to sit in one common coordinate system. We highly prioritize our efforts to make the resulting registered geometric model accurate compared with making the whole procedure computationally fast. For this reason, we design our registration procedure to be a simultaneous registration method based on an error metric computed from point-point distance, including additional attributes in its metric. Also, for robustness and efficiency, we adopt a conjugate gradient framework utilizing M-estimator to solve the least-square problem of minimizing the total errors through registration. Since we target large objects like the Great Buddha in Kamakura, the data size of each range image becomes huge. Thus, we employ k-d tree data structures for efficient point-point correspondence searches.

The remainder of this paper is organized as follows. In section 2, we overview related work and present our framework. Section 3 describes how a point correspondence search will be accomplished efficiently; and we describe the details of how least- square minimization of the objective function, the core of our simultaneous registration framework, in section 4. We show results of applying our approach to real data in section 5, and section 6 concludes the paper.

2 Overview

2.1 Related Work

Past work on range image registration can be roughly classified with respect to the following three aspects.

Strategy: simultaneous[1] or sequential The basic strategy of registering multiple range images can be represented by two different approaches. The straightforward strategy is to focus on only two range images at a time, and register each range image to another [25]. After one range image pair is registered, a new pair including either range image in the former pair, positioned in the resulting coordinate, is registered. This is repeated till all range images are used. Since this sequential strategy requires only two range images for each registration stage, it can be implemented with less memory and the overall computational cost tends to be cheap. Also, the computational cost for each registration stage is not affected by the number of range images to be

[1] Commonly refered to as "global registration" and "multi-view registration", especially in the graphics community

registered consequently.

However, this straightforward strategy is well known to be less accurate. In each range image pair registration stage, some error will be introduced due to data noise, etc. Since each range image will be fixed in the resulting position for each registration stage, this unavoidable error will be propagated to the latter registration stage and it will result in unaffordable error accumulated in the last range image position. Although the "gap" developed by this error accumulation can be small enough depending on the use of the resulting geometric model, it is much more preferable to avoid this theoretically, especially when the geometric model will be used as a basis of texture-mapping or appearance analysis, and so on.

Simultaneous registration solves this error accumulation problem by aligning all range images at once [1, 2, 5, 6, 8, 16, 20, 22, 23]. This can be accomplished by defining an error minimization problem by using an error metric common among all range images. This approach can diffusively distribute the registration error over all overlaps of each range image. The drawback is its large computational cost as opposed to that of sequential approaches.

Matching unit: features or points When registering range images, the problem is usually redesigned as an error (distance) minimization problem. The basis of the error to be measured can be features derived from the range images or points consisting of the range data. Feature-based methods extract some signatures around 3D points, invariant to Euclidean transformation, in each target range image and make correspondences among those features [6, 17, 18]. Based on the assumption that all correspondences are matched correctly, the transformation for registration can be computed in a closed form manner. On the other hand, if the signatures computed from the range images do not provide enough information and the matching of them cannot be done correctly, the registration stage can fail miserably. Point-based methods directly use the 3D points in an iterative manner. The point mates, the point correspondences to compute the error metric, are dynamically updated and several iterative steps are used to minimize the total error. One drawback of this point-based approach is that it requires an initial estimation of the rough transformation between the target range images, which is normally provided by human hand or interaction, while most feature-based approaches do not have this requirement.

Error metric: point-point distance or point-plane distance Originally, point-based approaches, such as the ICP algorithm [4, 28], set the error metric basis on the Euclidean distance between two points corresponding each other [10, 20]. However, since this error metric does not take the surface information into account, the point-based approaches based on point-point distance suffer from the inability to "slide" overlapping range images. An alternative to this distance metric is to use point-plane Euclidean distance, which can be computed by evaluating the distance between the point and its mate's tangent plane [6, 22]. By embedding the surface information into the error metric in this way, point-based approaches utilizing point-plane distance metric tend to be robust against local minima and converge quickly. However, computing the point-plane distance is computationally expensive compared with point-point distance computation; thus, methods using viewing direction to find the correspondence are also proposed for efficiency [1, 5, 22].

2.2 Our Approach

Taking into account the consideration described above, we have designed a registration algorithm which is i) based on the simultaneous strategy, ii) using points as matching units, iii) with the point-point distance metric. The framework is inspired by the work of Wheeler et al [26, 27], that applied similar techniques for object recognition and localization.

We want to construct the geometric model to be as accurate as possible. Also as future work, we would like to accomplish appearance analysis making considerable use of the geometry. For this reason, as a preliminary step, we attach more importance to robustness and accuracy than to computational expense in the registration method. This causes us to choose a simultaneous strategy, which is accurate in principle.

We employ points as matching units. Although the laser range scanner we use is quite accurate, still the distance to the object is large and the measurement condition is poor in many cases. Because the scanned range images include noise, the information computed from them will be even more corrupted by that noise. Thus, we avoid using any secondary features derived from raw range data; instead, we directly use data points as matching units.

We use the point-point distance metric. Due to the noise problem, as mentioned above, we have to avoid obtaining secondary features, surface normals in this case, and thus, cannot use the point-plane metric that requires us to calculate surface normals. It is also true that point- point metric is less expensive in computational cost than the point- plane metric, and is preferable when the data set is very large.

The overall simultaneous registration framework can be described as an iteration of the following procedure until it converges.

```
Procedure OneStepOfSimultaneousRegistration

Array KDTrees, Scenes, PointMates, Transforms

foreach r in AllRangeImages
  KDTrees[r] = BuildKDTree(r)

foreach r in AllRangeImages
```

```
  foreach s in AllRangeImage-r
    Scenes[s] = s

  foreach i in Pointsof(r)
    foreach s in Scenes
      PointMates[i] += CorrespondenceSearch(i,
KDTree[s])
  Transforms[r] = TransformationStep(PointMates)

TransformAll(AllRangeImages, Transforms)
```

We basically extend the framework of the pairwise ICP algorithm to handle multiple range images simultaneously. This is achieved by setting up an objective function to minimize globally, with respect to each of the range images. Defining *model* as the particular range image in interest and *scene* as one of the range images in the rest of range image set, in one simultaneous registration loop, each range image becomes a *model* once. Point mate search (search for nearest neighbor point) for each point in the *model* is done against all *scene* range images ($M-1$ if we have M range images), and they are stored in a global array. Rigid transformation for the current *model* is computed in a conjugate gradient search framework utilizing M-estimator, and is stored in a global array. After each range image has become a *model* once, all range images are transformed using the transformation stored in the global array. Note that each range image is not transformed immediately. Considering that each step transformation evaluated inside one simultaneous registration procedure will not be so large, this latency of transformation will not cause a problem. Furthermore, this timing of transformation saves us a large amount of computational time, since construction of k-d trees is required only once per range image in one simultaneous registration procedure. Details will be discussed in the following sections.

3 Point Mate Search

3.1 K-D Tree

As we try to register range images that consist of a large amount of 3D points, finding correspondences for each point in each range image can easily dominate a critical portion of the overall computational time. To obtain point correspondences efficiently, we employ k-d tree structure to store the range images [11]. K-d tree's k-d abbreviates *k-dimensional* and it is a generalization of a binary-search tree for efficient search in high dimension space. The k-d tree is created by recursively splitting a data set down the middle of its dimension of greatest variance. The splitting continues until the leaf nodes contain a small enough number of data points.

The k-d tree constructed becomes a tree of depth $O(\log N)$ where N is the number of points stored. A nearest-neighbor search can be accomplished by following the appropriate branches of the tree until a leaf node is reached. A hyper-sphere centered at the key point with a radius of the distance to the current closest point can be used to determine which, if any, neighboring leaf nodes in the k-d tree must be checked for closer points. Once have we tested all the data in leaf nodes which could possibly be closer, we are guaranteed to have found the closest point in the tree. Though its worst case complexity is $O(N)$, the expected number of operations for the nearest-neighbor search is $O(\log N)$, which will be the case if the data is evenly distributed. For the cases of storing surfaces in 3D space in k-d trees, usually this even distribution assumption holds. The largest overhead involved in using k-d trees is that the k-d tree of range-image points must be built prior to the search. This operation costs $O(N \log N)$. To avoid making this computational expense critical, we update each range image position only once in one simultaneous registration procedure as listed in the psuedo code in section 2.2, requiring only M times of k-d tree rebuilds in one global iteration, where M is the number of range images.

3.2 Distance Metric

To utilize a nearest-neighbor search based on k-d tree structure, we need a measure of dissimilarity between a pair of points. The dissimilarity, Δ, between k-d points $\mathbf{x}$ and $\mathbf{y}$ must have the form

$$\Delta(\mathbf{x}, \mathbf{y}) = F(\sum_{i=1}^{k} f_i(\mathbf{x}_i, \mathbf{y}_i)) \tag{1}$$

where the functions f_i are symmetric functions over a single dimension and functions f_i and F are monotonic. All distances satisfy these conditions, including the Euclidean distance $\|\mathbf{x}-\mathbf{y}\|$. As mentioned in section 2.1, using point-plane distance as the error metric provides faster convergence . However, the point-plane distance, which can be computed by

$$\Delta(\mathbf{x}, \mathbf{y}) = (\mathbf{x}-\mathbf{y}) \cdot \mathbf{N}_{\mathbf{y}} \tag{2}$$

does not satisfy the monotonic condition. To take advantage of the efficiency of the k-d tree structure, we use the point-point Euclidean distance as the dissimilarity measure. Also, we prefer point-point distance for the sake of robustness; avoiding the usage of secondary information derived from raw data, such as surface normals in point-plane, which can be sensitive to noise in the raw data points.

Figure 1 depicts an example of point correspondences in the case of using point-point distance metric and point-plane distance metric. While the point-point distance metric searches for the nearest neighboring point, meaning establishing a discrete mapping of one surface to another, the point-plane distance metric can be considered as a way to find the continuous mapping of one surface to another. In cases like Figure 1, where the *model* surface has to be "slid" to fit the *scene* surface, the point-plane approach succeeds in finding the correspondences that enable us to compute the rigid transformation close to the sliding direction, while the

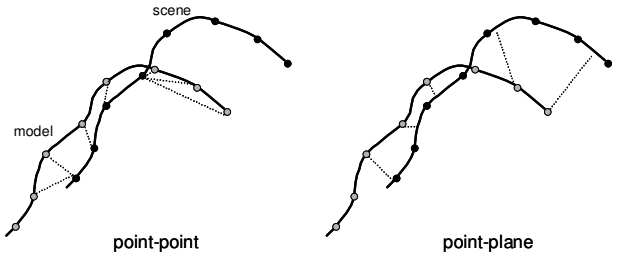

Figure 1. *Point correspondences using point-point and point-plane distance metric.*

Figure 2. *Images using laser reflectance strength as pixel values.*

point-point approach tends to get stuck in a local minima because of the inability to find point mates in the sliding direction. This sliding ability of point-plane approaches provides faster convergence compared with using point-point distance metric.

To compensate for the inability of sliding in point-point based distance measurement, we need to attach, to the 3D points, some information that suggests better matches. For this purpose, we use the laser reflectance strength value (referred to as RSV in the rest of this paper) as an attribute of each 3D point. Most laser range finders return the strength of the laser reflected at each surface point that it measured as an additional output value. Figure 2 shows two images with RSVs used as the pixel values. For better visualization, the images are histogram-equalized. As can be seen, the RSVs are mostly invariant against Euclidean transformation, since the dominant factor of the power of laser reflected at an object surface is its surface material. One common method to utilize two different sources of information in distance measurement, in this case the position distance and RSV distance, is to set up a combined metric, such as

$$\Delta(\mathbf{x},\mathbf{y}) = [(x_{\mathbf{x}}-x_{\mathbf{y}})^2+(y_{\mathbf{x}}-y_{\mathbf{y}})^2+(z_{\mathbf{x}}-z_{\mathbf{y}})^2+\lambda(r_{\mathbf{x}}-r_{\mathbf{y}})^2]^{\frac{1}{2}} \quad (3)$$

where r is RSV and λ is a constant scalar. However, this scalar introduces a tedious and ad hoc effort to determining

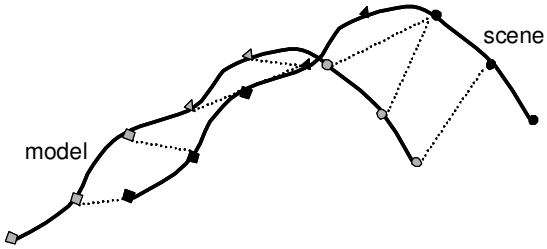

Figure 3. *Point mates using point-point distance metric with reflectance strength values. Different shape marks indicate different reflectance strength values.*

the "best" λ. Instead, we use the reflectance to determine the best pair among candidates of closest points. Namely, we first search for multiple (m) closest points in the k-d tree, and then evaluate the RSV distance for each of them to get the closest point with respect to laser reflectance strength value. We gradually reduce the number of the candidates m along the iteration so as to make it inversely proportional to the number of iterations. This utilization of the laser reflectance is similar to [19], which uses color attributes to narrow down the closest point candidates. Figure 3 depicts how the point-point distance metric utilizing RSV as additional attribute works in the example case depicted in Figure 1 ($m = 4$ in this example).

3.3 Speeding Up

Even though we employ k-d tree structure for efficient point correspondence search, when the number of points in the target range images get a large, the computational cost becomes massive. In early stages of the simultaneous registration, when the range images are widely distributed, it is more important to get them close to each other than to accurately compute the rigid transformation for each registration step. To provide a way to speed up the registration, we subsample each range image to reduce the number of points used in the registration process. The points in each range image are given a sequential identification number $m = 0, .., M-1$ and a uniformly distributed random number within the interval $[0, M-1]$ is generated to pick up the points to be used. The seed number to generate the random numbers is common for all range images in one simultaneous registration procedure and updated once per one global registration step. In the current implementation, we allow the user to determine the percentile of points to be used in each range image interactively. In future implementation, this could be done automatically by first using small percentage and gradually increasing it to reach one hundred percent.

As the range images are set to be still in one iteration of simultaneous registration, it is very easy to make the whole

framework run in a parallel manner. In our current implementation, constructing k-d trees and search points mates and computing transformation steps are done in threads, providing high scalability.

4 Least-square Minimization Strategy

4.1 Representing Transformation

Given a set of corresponding points $(\mathbf{x}_i, \mathbf{y}_i)$ where $i = 0, ..., N-1$, the registration problem is to compute the rigid transformation which registers the *model* points $\mathbf{x}_i$ with their corresponding *scene* points $\mathbf{y}_i$. The rigid transformation can be specified by a pair of a 3×3 rotation matrix $\mathbf{R}$ and a 3D translation vector $\mathbf{t}$. When the corresponding points are aligned with each other, $\mathbf{y}_i$ can be written as

$$\mathbf{y}_i = \mathbf{R}\mathbf{x}_i + \mathbf{t} \tag{4}$$

Since range data points will be contaminated by noise, the range image registration problem can be described as an error minimization problem with the error function as follows:

$$f(\mathbf{R}, \mathbf{t}) = \sum_i \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\|^2 \tag{5}$$

to minimize with regard to $(\mathbf{R}\ \mathbf{t})$. As mentioned in section 2.2, i will stand for all point mates established from all pairs of range images (if there are M range images, i will include all point mates from $M \times (M-1)$ range image pairs). Although it is convenient for vector computation to represent the rotation as a 3×3 matrix $\mathbf{R}$, $\mathbf{R}$ will be constrained in a non-linear way as follows (T stands for transpose):

$$\begin{aligned} \mathbf{R}\mathbf{R}^T &= \mathbf{I} \\ |\mathbf{R}| &= 1 \end{aligned}$$

It is difficult to take advantage of the linear matrix representation of rotation while satisfying these constraints. For this reason, we will use the quaternion representation for rotation, which is a well known solution to this rotation problem. (The benefits of using quaternion will be described later.) Thus, the position parameters of each range image and the rigid transformation to register all of them will be represented with seven element vectors as follows:

$$\begin{aligned} \mathbf{p} &= [\mathbf{t}^T\ \mathbf{q}^T]^T \\ where\ \ \mathbf{q} &= [u\ v\ w\ s]^T \end{aligned} \tag{6}$$

4.2 M-Estimator

As seen in section 4.1, the registration problem can be described as a least-square minimization problem with the objective function equation (5). Point correspondences are acquired using the techniques described in section 3. On solving this error minimization problem, we will have to deal with two problems,

Poor initial correspondences We must assume that the point correspondences established in the beginning will include a large number of mismatches.

Outliers Even when most of the point correspondences are correct, we still have to deal with outliers resulting from mismatches and noise-corrupted data points.

The underlying problem here is how to robustly reject outliers. The following three representative classes of solutions can be found in the field of robust statistics. The first class of solutions, outlier thresholding, is the simplest and most computationally cheap technique; thus it is the most common technique used in vision applications. The basic idea is to estimate the standard deviation σ of the errors in the data and to then eliminate data points which have errors larger than $|k\sigma|$ where k is typically greater than or equal to 3. The problem of outlier thresholding is that a hard threshold is determined to eliminate the outliers. This means that, regardless of where the threshold is chosen, some number of valid data points will be classified as outliers and some number of true outliers will be classified as valid. In this sense, it is unlikely that a perfect method for selecting the threshold exists unless the outliers are all known a priori.

The second class of robust estimators is the median/rank estimation method. The basic idea is to select the median or kth value (for some percentile k) with respect to the errors for each observation and to then use that value as our error estimate. The logic behind this is that the median is almost guaranteed not to be an outlier as long as half of the data is valid. An example of median estimators is the least-median-of-squares method (LMedS). LMedS computes the parameters of interest which minimize the median of the squared error computed from all data pairs using that parameter. Essentially, this requires an exhaustive search of possible values of the parameters by testing least-squares estimates using that parameter for all possible combinations of point correspondences. While these median-based techniques can be very robust, this exhaustive search remains a large drawback.

The third class of robust techniques is M-estimation; the technique we use. The general form of M-estimators allows us to define a probability distribution which can be maximized by minimizing a function of the form

$$E(z) = \sum_i \rho(z_i) \tag{7}$$

where $\rho(z)$ is an arbitrary function of the errors z_i in the data set. The M-estimate is the maximum-likelihood estimate of the probability distribution P equivalent to $E(z)$. Least-squares estimation, such as minimizing (5), corresponds to M-estimation with $\rho(z) = z^2$.

$$P(z) = e^{-E(z)} = e^{-\sum_i z_i^2} \tag{8}$$

We can find the parameters $\mathbf{p}$ that minimize E by taking the derivative of E with respect to $\mathbf{p}$ and setting it to 0.

$$\frac{\partial E}{\partial \mathbf{p}} = \sum_i \frac{\partial \rho}{\partial z_i} \cdot \frac{\partial z_i}{\partial \mathbf{p}} = \sum_i w(z_i) z_i \frac{\partial z_i}{\partial \mathbf{p}} = 0 \tag{9}$$

where $w(z) = \frac{1}{z}\frac{\partial \rho}{\partial z}$

As can be seen in (9), M-estimation can be interpreted as weighted-least square minimization, with the weight function w being a function of data points z_i. In our current implementation, we use the Lorentz function as the M-estimator because we found it to work best with our range image data.

4.3 Putting It Together

Now, we can redefine our registration problem as follows: Given a set of corresponding points $(\mathbf{x}_i, \mathbf{y}_i)$ (i=0,...,N-1), we will minimize

$$E(\mathbf{p}) = \frac{1}{N}\sum_{i}^{N}\rho(z_i(\mathbf{p})) \quad (10)$$

$$\textit{where } z_i(\mathbf{p}) = \|\mathbf{R}(\mathbf{q})\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\| \quad (11)$$

$$\textit{and } \rho(z_i) = \log(1 + \frac{1}{2}z_i^2) \quad (12)$$

The minimization of function E can be accomplished in a conjugate gradient search framework. Conjugate gradient search is a variation of gradient descent search; it constrains each gradient step to be conjugated to the former gradient step. This constraint avoids much of the zig-zagging that pure gradient descent will often suffer from, and consequently provides faster convergence.

In applying conjugate gradient search to our minimization problem, we need to compute the gradient of function E with respect to pose parameter $\mathbf{p}$ which can be described as equation (9). For the following derivations, we redefine z_i to be

$$z_i(\mathbf{p}) = \|\mathbf{R}(\mathbf{q}\mathbf{x}_i) + \mathbf{t} - \mathbf{y}_i\|^2 \quad (13)$$

A priori to the computation of the gradient, we pre-rotate the model points, so that the current quaternion is $\mathbf{q}_I = [0\ 0\ 0\ 1]^T$ which has the property that $\mathbf{R}(\mathbf{q_I}) = \mathbf{I}$. This allows us to take advantage of the fact that the gradient of $\mathbf{R}(\mathbf{q})\mathbf{x}$ can easily be evaluated at $\mathbf{q} = \mathbf{q}_I$:

$$\frac{\partial(\mathbf{R}\mathbf{x})}{\partial\mathbf{q}}\mathbf{x} = 2\mathbf{C}(\mathbf{x})^T b \quad (14)$$

where $\mathbf{C}(\mathbf{x})$ is the 3×3 skew-symmetric matrix of the vector $\mathbf{x}$ which has the useful characteristic as follows.

$$\mathbf{C}(\mathbf{x})\mathbf{y} = \mathbf{x} \times \mathbf{y} \quad (15)$$

where $\times$ is the cross product. With these facts, $\frac{\partial z_i}{\partial \mathbf{p}}$ in equation (9) can be derived as

$$\begin{aligned}\frac{\partial z_i}{\partial \mathbf{p}} &= 2(\mathbf{R}(\mathbf{q})\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i)\frac{\partial(\mathbf{R}(\mathbf{q})\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i)}{\partial\mathbf{p}} \\ &= \begin{bmatrix} 2(\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i) \\ 4\mathbf{C}(\mathbf{x})^T(\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i)\end{bmatrix} \\ &= \begin{bmatrix} 2(\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i) \\ 4\mathbf{x}_i \times (\mathbf{t} - \mathbf{y}_i)\end{bmatrix} \end{aligned} \quad (16)$$

With the gradient computed in the above manner, line minimization is accomplished with a golden section search. Line minimization methods using interpolation are not adopted, since it is easy to imagine the base function to be highly non-linear.

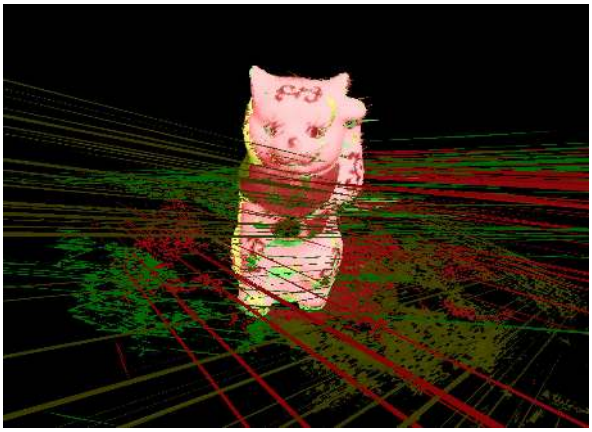

Figure 4. *Initial positions of the Noisy Cat sequence.*

5 Results

5.1 Noisy Range Images

To examine its robustness against noise, we applied the proposed method to a noisy range image sequence. We built a light stripe range finder [24], and scanned a ceramic cat. By setting the threshold of the light stripe range finder to include quite an amount of background and not to eliminate ill triangle patches (triangle patches that have large aspect ratios), we obtained three range images including a lot of noise. To compare the proposed method with the registration method proposed in [22] [2], the range images were initially aligned with each other by means of human interaction as depicted in Figure 4. [3] After iterating both methods until convergence, we eliminated all 3D points and triangle patches that did not belong to the ceramic cat and measured the errors by using a point-plane distance metric. Table 1 shows the results and Figure 5 depicts the histograms of errors for both methods. Our method converged robustly, while the method of [22] converged into a local minima, leaving a gap as can be seen in Figure 6.

	Average Error	Max. Error	Min. Error
Our method	0.84	2.55	5.35×10^{-7}
[22]	1.29	2.57	3.21×10^{-5}

Table 1. *Comparison of errors in mm.*

5.2 Preserving Cultural Heritage Objects

We have applied the proposed method to register real data, the Great Buddha in Kamakura (Figure 7): a 13m tall statue sitting in open air. The Great Buddha was scanned from fourteen different directions using Cyrax 2400 [14], a

[2] We implemented the registration method in [22] based on the paper, meaning the comparison may not be fair.

[3] With more rough initial hand alignment, the other registration method did not converge.

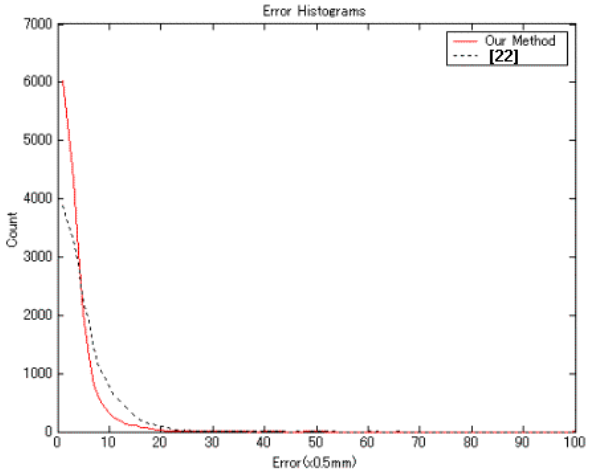

Figure 5. *Histogram of errors.*

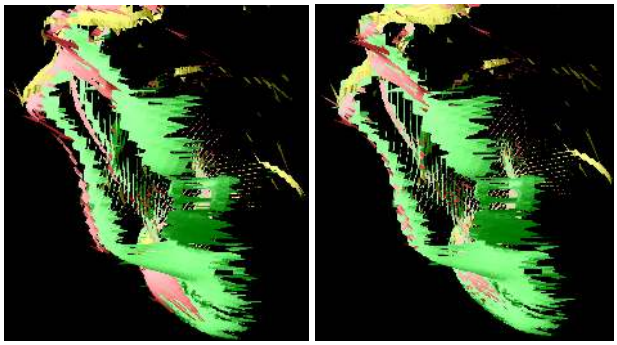

Figure 6. *Left: Registered with [22] Right: Registered with our method.(Viewing from the top left of the cat.)*

Figure 7. *A photograph of the Great Buddha in Kamakura City.*

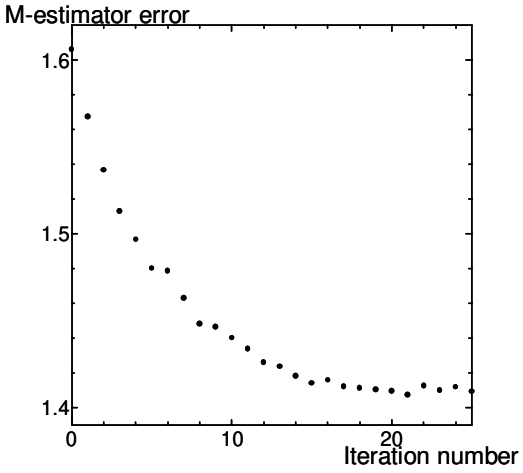

Figure 8. *M-estimator error v.s. iteration number*

time-of-flight laser range scanner that can scan up to 100m with $\pm$6mm error at 50m distance. Each point cloud image consists of approximately three to four million vertices. Since registering all range images with full resolution requires massive computational resource and time, we registered those range images in $1/25$ resolution as a preliminary experiment.

First the input range images were registered in a pairwise manner with occasional human interaction for initial alignment; they were then registered simultaneously. The variance of Lorentz's function was set large in the beginning and then gradually decreased each time the registration procedure converged with a particular variance value. Rough initial pairwise alignment was accomplished with around five to ten iterations, and the final simultaneous registration was done with 25 iterations. Figure 8 depicts the M-estimator error for each iteration for the last 25 iteration. Since all range images are treated to be static inside each iteration, the M-estimator error does not always get smaller after each iteration compared with the former iteration. However, because the error is guaranteed to decrease inside each iteration, it is clear that the algorithm converges to a certain minimum which is shown in the graph.

Figure 9 shows the resulting Great Buddha rendered as a point cloud.

6 Conclusion and Future Work

We have proposed a framework to simultaneously register multiple range images. The simultaneous registration problem is redefined as a least- square problem with an objective function globally constructed with respect to each range image. For efficiency, we employ k-d tree structure for fast point correspondence search and apply conjugate gradient search in minimizing the least-square problem for faster convergence. For robustness, we employ the laser re-

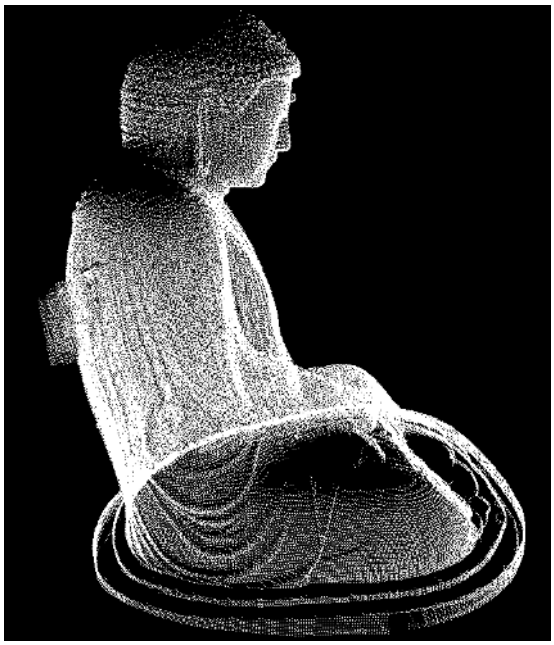

Figure 9. *Registered Great Buddha.*

flectance strength as an additional attribute of the 3D points and search for "better" point mates based on their distance. Also, M-estimator is used for robust outlier rejection.

For future work, we plan to automate initial estimation of the rigid transformations to pass to the simultaneous registration program, which is currently done by human interaction.

Acknowledgement

This work was supported by Ikeuchi CREST (Core Research for Evolutional Science and Technology) of Japan Science and Technology Corporation.

References

[1] R. Benjemma and F. Schmitt. Fast global registration of 3d sampled surfaces using a multi-z-buffer technique. In *Proc. Int. Conf. On Recent Advances in 3-D Digital Imaging and Modeling*, pages 113–120, May 1997.

[2] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Trans. Patt. Anal. Machine Intell.*, 18(5):540–547, May 1996.

[3] R. Bernardini and H. Rushmeier. The 3d model acquisition pipeline. In *Eurographics 2000 State of the Art Report (STAR)*, Aug. 2000.

[4] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(2):239–256, Feb 1992.

[5] G. Blais and M.D. Levine. Registering multiview range data to create 3d computer objects. *IEEE Trans. Patt. Anal. Machine Intell.*, 17(8):820–824, Aug 1995.

[6] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, Apr 1992.

[7] C. Dorai, G. Wang, A.K. Jain, and C. Mercer. From images to models: Automatic 3d object model construction from multiple views. In *Proc. of the 13th IAPR International Conference on Pattern Recognition*, pages 770–774, 1996.

[8] D.W. Eggert, A.W. Fitzgibbon, and R.B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering. Technical Report 804, Dept. of Artificial Intelligence, University of Edinburgh, 1996.

[9] M. Levoy et al. The digital michelangelo project: 3d scanning of large statues. In *ACM SIGGRAPH 2001*, pages 131–144, Jul. 2001.

[10] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *International Journal of Robotic Research*, 5(3):27–52, Fall 1986.

[11] J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. On Mathematical Software*, 3(3):209–226, 1997.

[12] G. Godin, M. Rioux, and R. Baribeau. Three-dimensional registration using range and intensity information. In *Proc. SPIE vol.2350: Visionmetrics III*, pages 279–290, 1994.

[13] http://www.cyberware.com.

[14] http://www.cyra.com.

[15] http://www.minolta rio.com/vivid/.

[16] H. Jin, T. Duchamp, H. Hoppe, J.A. McDonald, K. Pulli, and W. Stuetzle. Surface reconstruction from misregistered data. In *Proc. SPIE vol.2573: Vision Geometry IV*, pages 324–328, 1995.

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

[18] A. Johnson and M. Hebert. Surface registration by matching oriented points. In *Proc. Int. Conf. On Recent Advances in 3-D Digital Imaging and Modeling*, pages 121–128, May 1997.

[19] A. Johnson and S.B. Kang. Registration and integration of textured 3-d data. In *Proc. Int. Conf. On Recent Advances in 3-D Digital Imaging and Modeling*, pages 234–241, May 1997.

[20] T. Masuda, K. Sakaue, and N. Yokoya. Registration and integration of multiple range images for 3-d models construction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 879–883, Jun 1996.

[21] D. Miyazaki, T. Oishi, T. Nishikawa, R. Sagawa, K. Nishino, T. Tomomatsu, Y. Takase, and K. Ikeuchi. The great buddha project: Modelling cultural heritage through observation. In *6th International Conference on Virtual Systems and MultiMedia VSMM2000*, pages 138–145, Oct. 2000.

[22] P. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Proc. Int. Conf. on Shape Modeling and Application*, pages 130–139, Mar 1997.

[23] K. Pulli. Multiview registration for large data sets. In *Second Int. Conf. on 3D Digital Imaging and Modeling*, pages 160–168, Oct 1999.

[24] K. Sato and S. Inokuchi. Range-imaging system utilizing nematic liquid crystal mask. In *First International Conference on Computer Vision*, pages 657–661, 1987.

[25] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH 94*, pages 311–318, Jul 1994.

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

[27] M.D. Wheeler and K. Ikeuchi. Sensor modeling, probablistic hypothesis generation, and robust localization for object recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(3):252–265, 1995.

[28] Z. Zhang. Iterative point matching for registration of free form curves and surfaces. *International Journal of Computer Vision*, 12(2):119–152, 1994.