High-resolution Surface Reconstruction based on Multi-level Implicit Surface from Multiple Range Images

SHOHEI NOGUCHI1,a) YOSHIHiro Watanabe1 MASatoshi Ishikawa1

Received: October 10, 2012, Accepted: May 23, 2013, Released: August 23, 2013

Abstract: Sensing the 3D shape of a dynamic scene is not a trivial problem, but it is useful for various applications. Recently, sensing systems have been improved and are now capable of high sampling rates. However, particularly for dynamic scenes, there is a limit to improving the resolution at high sampling rates. In this paper, we present a method for improving the resolution of a 3D shape reconstructed from multiple range images acquired from a moving target. In our approach, the alignment and surface estimation problems are solved in a simultaneous estimation framework. Together with the use of an adaptive multi-level implicit surface for shape representation, this allows us to calculate the alignment by using shape features and surface estimation according to the amount of movement of the point clouds for each range image. By doing so, this approach realized simultaneous estimation more precisely than a scheme involving mere alternating estimation of shape and alignment. We present results of experiments for evaluating the reconstruction accuracy with different point cloud densities and noise levels.

Keywords: range image, 3D point cloud, alignment, surface reconstruction, resolution enhancement

1. Introduction

Sensing and modeling of the three-dimensional (3D) shape of a dynamic scene is an important technique that can be widely used in the fields of robotics, video, interfaces, inspection, and numerical simulation. However, existing 3D shape-acquisition technology has been designed mainly for stationary objects and is not suitable for moving objects. In addition, it has been difficult to acquire sequential multiple range images at a high sampling rate. In contrast, a 955-fps real-time shape-measurement vision system based on a structured light method and capable of high-frame-rate imaging of moving objects and parallel image processing has been developed [1]. However, the resolution of this measurement system is limited.

Therefore, here we propose a reconstruction method for enhancing the resolution of the surface of an object after obtaining a 3D video. In this method, the measurement target is a moving rigid object, and we use multiple range images. Our method has no constraints in terms of the target motion. For each range image, the points were sampled at different positions on the target surface depending on the relative position of the sensor and the target object. Figure 1 illustrates the measurement environment for this method. We use active measurement techniques like the structured-light method to measure multiple range images. In addition, the proposed method can be applied to dynamically changing situations, such as when the sensor system is moving.

In the case of acquiring a continuous 3D video, the proposed method is useful for a variety of applications because it reconstructs high-quality data beyond the capacity of the sensing system itself. Figure 2 shows some possible new applications that have never been achieved before. As shown in the figure, a high-resolution surface can be acquired from a self-propelled object that simply goes through the sensor. In addition, by utilizing free movement of the sensor, like throwing or rolling it, the shape of a large object that is impossible to measure with a conventional...
sensor system can also be obtained at high resolution. A similar sensing system has been proposed [2], although the system is not designed for 3D sensing.

In order to realize reconstruction using a sparse point cloud, the proposed method controls the local weight for the estimation based on the complexity of the shape. This control makes it possible to adaptively estimate the motion with a simple region of the surface and contributes to improved accuracy, making the method more robust for sparse point clouds. In addition, by estimating the surface normal, which is important information about the shape, our method avoids errors in motion estimation which can easily occur with sparse input points (e.g., sensing a similar shape but with the front and back reversed).

2. Background

Our task involves alignment, shape estimation, and normal estimation. However, these three aspects have been studied separately in related research on range image processing, and there are very few methods incorporating all three tasks.

With respect to alignment of range images, conventional methods mainly focus on the alignment of two high-resolution range images [3], [4]. In particular, the Iterative Closest Point (ICP) method is widely used [3]. However, when using this method for multiple range images, the error would be cumulative, reducing the accuracy, because this method assumes only two range images as inputs. Some studies using multiple range images as inputs have been reported. For example, in order to conserve the details in the 3D shapes of culturally important objects, there is a method that minimizes a single global error function constructed from multiple range images [5], [6]. However, in these methods, a low-resolution range image is not assumed. In sparse input data, it is difficult to find corresponding points, which limits the alignment accuracy. In order to properly solve the problem of reconstructing a high-resolution surface from unaligned multiple range images while ensuring robustness even for sparse inputs, it is not enough to focus only on the alignment scheme. It is important to construct a new method that combines alignment and surface estimation in a complementary style.

Next, we describe a method of representing the shape. Mesh-based surface representation has a high degree of freedom and can produce a faithful representation of the shape, but has the disadvantage that the amount of data needed is high. Also, NURBS and Bezier surfaces are typical parametric surfaces used to represent shapes. However, the shapes that can be represented by them are limited, and free shape representation is difficult.

Recently, many shape reconstruction methods using the implicit surface representation have been proposed. These include, for example, the Radial Basis Function (RBF) [7], a method combining Partition of Unity (PU) and the RBF method [8], and Multi-level Partition of Unity (MPU) [9]. These representations have a high degree of freedom and require a small set of parameters. Also, the method in Ref. [10] shows that this implicit surface representation enables a sophisticated formulation for motion estimation. In this paper, we also focus on the feature that an implicit surface is also useful for normal estimation and noise removal. We employed MPU because it can represent complex shape features and can control the fitting accuracy hierarchically.

In these methods of reconstructing the shape, surface-normal information is important. In general, however, a sensor designed for measuring a 3D point cloud with high accuracy is not good at measuring the normal and the points simultaneously. In contrast, there are some methods that can estimate the normal from a point cloud. The method proposed by Hoppe et al. [11] is commonly used. However, when the input is a sparse point cloud, the estimated normal is unreliable. In our method, the accuracy of normal estimation is improved by introducing an updating scheme combining motion estimation and surface estimation.

Some unified methods involving alignment and surface estimation have been reported. Fuhrmann et al. proposed a method that integrates multiple range images of different resolutions to reconstruct the shape [12]. This method represents the shape with an implicit surface and uses input data obtained by a stereo camera. However, the resolution cannot be improved from that of the original input data because the input shape is obtained from stereo matching. There is another method that assumes multiple range images as inputs [13], [14]. This method uses a signed distance field and estimates motion parameters and a mesh surface in a unified framework. However, the goal of these methods is not to enhance the resolution, and the data used was high-resolution data.

Furthermore, similarly to our approach described in this paper, a method using simultaneous estimation to solve the problem of reconstructing a high-resolution surface has also been proposed [10]. It is a compact method that utilizes the characteristics of implicit surfaces; however, there is a problem in terms of conflict between the alignment process and the shape estimation process in the simultaneous estimation framework. In particular, it is difficult for the method to reconstruct a complex shape, because the implicit surface used is based on RBF, and an implicit surface cannot sufficiently represent local sharp features. In addition, all input points were treated equally, and an adaptive scheme that reduces the effects of estimation and measurement errors was not introduced.
3. Surface Reconstruction Based on Adaptive Multi-level Implicit Surface

3.1 Outline

In the proposed method, we use multiple low-resolution range images as inputs and reconstruct a high-resolution surface from them. The target object is a rigid body.

Figure 3 illustrates an outline of this method. We consider $N$ range images as the input and regard a measured range image as a point cloud described by $P = \{ p_f : f = 1, \ldots, N \}$. Here, subscript $f$ stands for the frame number. In addition, the position of the sensor system when each frame is measured is described as $O = \{ o_f : f = 1, \ldots, N \}$ and $o_f \in \mathbb{R}^{3 \times 1}$. Input range images have different coordinate systems because the target object is moving. For this reason, they have to be aligned to a common coordinate system. In this case, we call the frame that serves as the basis for alignment of each frame the base frame, and we call the coordinates in this frame the base coordinates. In this paper, we set the $N/2$th frame as the base frame.

A continuous curved surface shape is represented by the shape parameters $S$, and alignment to the base coordinates of the range image is represented by motion parameters $M = \{ m_f : f = 1, \ldots, N \}$.

Under the definitions above, the reconstruction is the problem of estimating shape parameters $S$ and motion parameters $M$ from point clouds $P$. Note that motion estimation and shape estimation are dependent on each other. Therefore, this task must be solved in a simultaneous estimation framework. We solve this simultaneous estimation approximately by repeating three processes in order: motion estimation, normal estimation, and shape estimation.

As mentioned above, we assume a sparse point cloud as an input. The features of our method for solving this problem are summarized below:

(I) Surface representation based on an implicit function
The surface representation we used enables effective interpolation of a point cloud that has holes. Also, it is easy to represent the surface with a small set of parameters. In addition, motion and normals can be estimated reasonably well by using the implicit function.

(II) Surface reconstruction using reliable aligned data
The point clouds of every frame are aligned with different levels of accuracy. Therefore, surface estimation is required to preferentially use reliable frames that are accurately aligned.

(III) Iterative normal estimation
Normals estimated in the initial cycle are unreliable because the input data is sparse. Our method solves this problem by introducing an iterative update scheme. Also, the estimated normals serve as powerful information for improving the accuracy of motion and surface estimation.

(IV) Motion estimation focusing on a region with a simple shape
A region with simple features tends to be estimated with high accuracy. Therefore, such regions should be aligned preferentially. Our method introduces a technique for controlling the alignment of simple regions, which makes the motion estimation much more robust to surface estimation errors.

(V) Effective motion estimation using normals of the point cloud
Our method estimates the motion such that the normals are consistent between the surface and the point cloud. This makes the motion estimation in our method more robust than a simple minimization problem in which the distances between the surface and the points are evaluated.

In particular, we consider that features (II) and (IV) described above are the most significant contributions. These two features complement each other and reduce the estimation error that occurs when using a sparse point cloud as an input. As a result, our method can reconstruct a high-resolution surface much more quickly and with high accuracy.

The flow of the proposed method is shown below. Here, $k$ stands for the cycle number, and subscript $j$ stands for a point in the point cloud of frame $f$.

1. **Initial Surface Estimation.** Estimate surface using base frame.
2. **Range Image Addition.** Add 20 frames, if there are range images that can be added.
3. **Motion Estimation.** Estimate motion parameters $m_f^{(k)}$ for each frame by using shape parameters $S^{(k-1)}$.
4. **Normal Estimation.** Estimate normals for the point cloud aligned by using motion parameters $m_f^{(k)}$.
5. **Shape Estimation.** Estimate shape parameters $S^{(k)}$ from the normals $n_f^{(k)}$ and aligned points $P_f^{(k)}$.
6. **Decide whether to end Motion Estimation.** End the motion estimation for frame $f$ if the average amount of movement of frame $f$ is less than a prescribed constant.

![Fig. 3 Outline of the proposed method.](image-url)
7. Repeat. Repeat steps 2 to 6 until motion estimation is completed for each frame.

8. Last surface estimation. Estimate the surface after the last cycle with a strict fitting parameter.

Here, \( P_{f,j}^{(k)} \) is the \( j \)-th 3D point belonging to frame \( f \) of cycle \( k \), and \( n_{f,j}^{(k)} \) is the normal of point \( P_{f,j}^{(k)} \). To achieve stable estimation, we add 2\( \delta_f \) frames and estimated the motion only for those added frames at the frame adding cycle. 2\( \delta_f \) frames is added evenly before and after around the base frame. At the non-adding cycle, we estimate motion for all frames. Details of the difference between this adding cycle and a non-adding cycle are described in the following sections.

### 3.2 Adaptive Multi-level Implicit Surface Estimation

There are several conditions for shape representation in this task. The number of estimated variables should be small, it should be possible to represent a high-degree-of-freedom surface, local smooth characteristics should be controllable, extraction of shape features should be possible, and fitting accuracy should be controllable according to the fitting error for each point. We use the implicit surface representation, which meets these requirements. From a number of possible methods that are available, we use MPU [9] as a base, which estimates a shape by using an adaptive multi-level implicit surface. MPU is suitable when there is missing data and non-uniform data. We think that this is also valid for our task, in which noise is not uniform on the surface, and input data is sparse.

The basic format of this implicit surface function is

\[
F(x) = \sum_i \varphi_i(x) Q_i(x). \tag{1}
\]

In this function, \( \varphi_i(x) \) is called the Partion of Unity (PU) function, and \( Q_i(x) \) is called the local shape function. The PU function must satisfy the following equation in a closed region \( \Omega \) of Euclidean space:

\[
\sum_i \varphi_i(x) \equiv 1, \quad x \in \Omega. \tag{2}
\]

The PU function \( \varphi_i(x) \) can be generated by:

\[
\varphi_i(x) = \frac{H_i(x)}{\sum_j H_j(x)}. \tag{3}
\]

When the MPU method is used to find an approximation of a curved surface, B-spline functions are known to be used for \( H(p_i) \), and we use the same function in the method proposed here. Figure 4 shows an overview of the surface estimation. The figure is simplified to two-dimensional form. As shown in the figure, the red, blue, and green dots represent the input data points. The rectangular area represents a subdivided cell. The orange dotted line represents the local shape function \( Q_i(x) \) for points included in each cell. The purple line represents an implicit surface function \( F(x) \) estimated from the data points.

In this method, the degree of fitting of the local shape function to the point cloud is adjusted depending on the parameter \( m_f \) estimated in the motion estimation process. More specifically, we assume that the smaller the estimated motion is, the higher the degree of convergence for the estimation of the corresponding local surface is, and so the points are trusted.

The following shows the detailed procedure for the shape estimation. First, set a bounded box that includes all of the input points. Next, divide the cell using octree and estimate the local shape function for each cell. If the value of the evaluation function \( e_i \) is more than the threshold \( \varepsilon \), then divide the cell again. We set the radius of the support sphere to \( \alpha d_i \), and use the set of points \( P_i \) within the sphere to estimate the local shape function in the \( i \)-th cell. Here, \( d_i \) represents the length of the diagonal of cell \( i \), and \( \alpha \) is an arbitrary constant. In addition, for the local shape function, we used the following bivariate quadratic polynomial:

\[
Q(x) = w - t^T A t. \tag{4}
\]

Here, \( t = [u, v, 1]^T \) and \((u, v, w)\) are local coordinates of the cell that corresponds to \( x \). The origin of these coordinates is the center \( c_i \) of the cell, and the positive direction of \( w \) is coincided with unit normal vectors \( n_c \), which is calculated from the normalized weighted arithmetic mean of the normals of points \( P_i \). The \((u, v)\) plane is orthogonal to \( n_c \). The weight function we used for this is the B-spline. \( A \in \mathbb{R}^{3\times3} \) is a symmetric matrix. In this case, the unknown factor \( A \) is determined by solving the following minimization problem:

\[
\hat{A} = \arg\min_{A} \sum_{p \in P_i} H(p_j)^2 Q(p_j)^2. \tag{5}
\]

Here, \( p_j \) refers to a point that is included in the set of points \( P_i \).

By using only a bivariate quadratic polynomial for the local shape function of the adaptive multi-level implicit surface, the size of the cell including point \( x \) is an indicator of the complexity at that point. Then, in the proposed method, we regard this indicator as a characteristic of the shape and use it in the process of motion estimation described in Section 3.4.

As an evaluation function for determining whether to subdivide the cell, we use a distance measure called the weighted Taubin
distance, given by:
\[
e_i = \max_{\|p^{(k)}_{f,j} - p^{(0)}_{f,j}\| < \text{tol}} \exp \left(-\frac{t_f}{\ell_n} \frac{Q(p^{(k)}_{f,j})}{\nabla Q(p^{(k)}_{f,j})} \right)
\]

When \(e_i\) is smaller than the user-specified smoothing parameter \(\varepsilon_0\), cell subdivision stops. Here, \(t_f\) is an average amount of movement estimated in the previous motion estimation process for point \(p^{(k)}_{f,j}\). We will give a detailed definition of this in Section 3.4. The exponential term is a weight function that we newly added. As mentioned above, because frames whose average amount of movement is small are expected to be close to the optimal solution in motion estimation, those frames are considered to be reliable. Therefore, strict fitting to the point belonging to such frames is considered to be valid.

At the cycle where range images are added, we set \(t_f = 0\) and changed the evaluation function to the unweighted Taubin distance.

3.3 Normal Estimation by Using Implicit Surface

Shape estimation in the proposed method needs information about the normal for each point in the range image. Therefore, the proposed method calculates the normals from the point cloud. First, we calculated the normal \(n^{(0)}_{f,j}\) for each point by using the method of Hoppe et al. [11]. However, the normals calculated by this method had uncertainty. In that method, the reversal of the normal was determined by Graph-cut theory. However, for sparse point cloud data, it may not be possible to calculate the exact value of the normal. Therefore, in this method, the problem of reversal of the normal was resolved by using the following equation:

\[
n^{(0)}_{f,j} = \begin{cases} -n^{(0)}_{f,j}, & \text{if } n^{(0)}_{f,j} \cdot (p^{(0)}_{f,j} - \delta_j) < 0 \\ n^{(0)}_{f,j}, & \text{else} \end{cases}
\]

By using this equation, the estimation error of the normal can be limited to \(\pi/2\) rad.

Next, the following describes the updating of the normal from the estimated normal in the initial step. Figure 5 shows the normal updating process. The accuracy of a normal calculated only from input range images is low, due to the limited resolution of the point cloud. Therefore, we use a normal calculated from the estimated surface \(S\) to update the normal at each point. We can assume that the point cloud is close enough to the estimated surface. Under this assumption, the normal \(n_S(p^{(k)}_{f,j})\) calculated from surface \(S\) can be approximated by the following equation because \(Q(p^{(k)}_{f,j}) = 0\):

\[
n_S(p^{(k)}_{f,j}) = \frac{\nabla F(p^{(k)}_{f,j})}{\|\nabla F(p^{(k)}_{f,j})\|} \approx \frac{\sum_j \varphi_j(p^{(k)}_{f,j}) \nabla Q_j(p^{(k)}_{f,j})}{\sum_j \varphi_j(p^{(k)}_{f,j}) \nabla Q_j(p^{(k)}_{f,j})}
\]

In our method, if Eq. (9) is satisfied, the normal is updated using Eq. (10):

\[
n^{(k+1)}_{f,j} = n^{(k)}_{f,j} + n_S(p^{(k)}_{f,j}) - T_f
\]

3.4 Motion Estimation Based on Shape Features

In motion estimation, the estimated target surface to be aligned is fixed. The movement of the \(j\)-th point of frame \(f\) is described by the following equation:

\[
P^{(k)}_{j,f} = R^{(k)}_f P^{(k-1)}_{j,f} + T^{(k)}_f
\]

where \(m^{(k)}_f = [R^{(k)}_f, T^{(k)}_f]\) is a parameter with six degrees of freedom, \(R^{(k)}_f\) is the rotational component of frame \(f\), and \(T^{(k)}_f\) is the translational component of frame \(f\). In the proposed method, \(N\) range images are estimated separately in this motion estimation process.

Here, we consider two types of regions: a region with complex shape features and a region with simple shape features. In the region with complex shape features, it is difficult to determine whether the scatter points are caused by estimation errors because the target object itself has very complex shape features. On the other hand, the region with simple shape features is expected to be less susceptible to aliasing, and the target object is thus expected to have similar simple shape features. Therefore, for motion estimation, it is necessary to devise a suitable approach, such as focusing on a region with simple shape features and estimating motion mainly using that region.

Figure 6 shows the concept of motion estimation. In the proposed method, for each frame, motion estimation is performed by solving the following constrained nonlinear optimization problem:

\[
\begin{align*}
\min_{m^{(k)}_i} & \sum_{i=1}^{D_f} \left[ W_s(p^{(k)}_{j,f}) B(p^{(k)}_{j,f})^2 + G(n^{(k)}_{i,j}) \right] \\
\text{s.t.} & \quad i^{(k)}_{j,f} < i^{(k-1)}_{j,f} 
\end{align*}
\]
where function $B$ represents the distance between the surface and a point, and function $G$ represent the degree of similarity of the normal calculated from the surface and the normal linked to the point estimated at the $k$-th cycle. These are given by the following equations:

$$
B(p^{(k)}_{f,j}) = \left| F(p^{(k)}_{f,j}) \right| / \| \nabla F(p^{(k)}_{f,j}) \| \quad (15)
$$

$$
G(n^{(k)}_{f,j}) = \frac{r^2}{2} (1 - n^{(k)}_{f,j} \cdot n_S(p^{(k)}_{f,j})). \quad (16)
$$

In addition, $D_f$ is the number of points belonging to frame $f$, $W_S$ is the weight function for motion estimation, $\epsilon_0$ is the smoothing parameter described in Section 3.2, $n_S$ is the normal approximation in Eq. (8), and $t_f$ is the average amount of movement, given by

$$
t_f^{(k)} = \frac{1}{D_f} \sum_{j=1}^{D_f} \| p^{(k)}_{f,j} - p^{(k-1)}_{f,j} \|. \quad (17)
$$

The weight function used in the proposed method is

$$
W_S(p_{f,i}) = b \left( \log_2(d_i/d_{\text{min}}) \right) / \log_2(d_0/d_{\text{minAll}}). \quad (18)
$$

From test results obtained in the preliminary experiment, we selected a B-spline function. In total, we tested three functions, including a B-spline function, a Gaussian function, and a quadratic polynomial function. These functions were selected because of their high smoothness. $d_{\text{min}}$ is the shortest diagonal length of the cell containing point $p_{f,i}$, and $d_{\text{minAll}}$ is the shortest of all diagonal lengths. $d_0$ is the diagonal length of the initial bounding box set as described in the previous section. Since the cell diagonal length becomes one half at the next level down, the hierarchy level could be calculated by using the logarithm of $d_{\text{min}}$ normalized by $d_0$. In addition, as mentioned above, in the motion estimation, it is necessary to focus on regions with simple shape features and to perform alignment mainly with those regions.

From the above-described property that the complexity of the shape and the size of the cell are related to each other due to the adaptive hierarchical structure, the weight function Eq. (18) satisfies the condition that the alignment task must focus on regions with simple shape features.

We set a weight function $W_S = 1$ and minimize the unconstrained nonlinear Eq. (13) until all of the input range images have been added. This is to achieve uniform alignment of the point cloud, because the shape estimated using only the base frame might be affected by aliasing, and it is expected that the estimated surface would lose its original shape features.

When the input range image is added, the overlap area between the estimated surface and the point can be considered to be small. To achieve robust estimation even if the overlap area is small, we adopt the RANdom SAmple Consensus (RANSAC) method [15] in the motion estimation process only for the cycle where range images are added. The details are as follows. First, a given number of points are randomly selected from $f$-th frame. Second, motion estimation is performed using these points. Third, all 3D points in $f$-th frame are moved using the estimated motion. Fourth, we count the number of points whose error calculated from the implicit surface is less than a threshold. Fifth, these steps are repeated, and we choose the motion with the lowest error as the motion of $f$-th frame.

4. Experiments

4.1 Accuracy Evaluation for Range Images with Various Resolutions and Noise Levels

In this section, we describe how we evaluated the effectiveness of the method when applied to range images with several resolutions and noise levels. We also compared the method with a method in which point clouds were aligned by ICP and surfaces were reconstructed by MPU. The target was a Stanford Bunny [16].

Figure 7 shows a rendering of the original image of the object. The number of mesh triangles that make up the Stanford Bunny was 69,451, and the size of the object was $78 \times 77 \times 60$ (original units: mm). The experiments described in this section and the next section were performed by using synthetic data.

Figure 8 shows the simulation conditions. We applied the proposed method to range images with five different resolutions that were obtained by varying the density of the structured light pattern. The movement per frame was set by setting the axis of rotation to $\theta = (2.1, 2)$, the angle of rotation to $\theta = 0.03$ rad, and the translation to $v = (0.2, -0.1, 0.3)$. The total number of frames was 30, and the 15th frame was set as the base frame. The total numbers of points measured in 30 frames at the five resolutions were 1,726, 2,716, 3,525, 4,776, and 6,981.

We also evaluated the method using noisy data. In this experiment, two types of noise were randomly added to the 3,525-point dataset. The noise was a vector having a random norm with a

---

© 2013 Information Processing Society of Japan
maximum of 1.0 and a vector having a random norm with a maximum of 0.5 in random directions.

In addition, in range image addition, $\delta_f = 5$. In the motion estimation using RANSAC, the number of points was 10, and the process was repeated 30 times. In this experiment, we set the ratio of the support radius to $\alpha = 1.25$. In range image addition, we set $\epsilon_0 = 0.04$, and after that we set $\epsilon_0 = 0.02$. For each frame, when the average of the amount of movement became less than 0.001, the motion estimation was ended. The ICP method was also repeated 10 times.

For the evaluation, we sampled the surface of the ideal model at regular intervals and represented the sampling points as $q_i (i = 1, \ldots, 17,430)$. At this time, at the same sampling points, we evaluated the effectiveness of the method by calculating the average and variance of the distance function $B(q_i)$ calculated from the estimated surface.

Figure 9 shows the evaluation results with several resolutions. The blue line is the result for the surface reconstructed using the base frame, and the red line is the result for the surface reconstructed using our method. At all resolutions, the surface reconstructed by the proposed method was better than that constructed using a single range image. In addition, the results when noise was added to the range images are also shown in Fig. 9. Purple and orange marks show the results for the surfaces reconstructed using noisy data with the two different noise levels. For random noise with a magnitude of about 1.5% relative to the size of the shape, our method achieved robust shape reconstruction.

We also performed experiments evaluating the performance of the conventional method. As the conventional method, we used ICP, which is the official function implemented in the Point Cloud Library [17]. We evaluated the performance using a dataset including different resolutions. The green line in Fig. 9 shows the result for the surface reconstructed by ICP. As shown in Fig. 9, with ICP, it was difficult to reconstruct the shape for the sparse inputs (50–200 points/frame). The surface reconstructed using ICP is shown in Fig. 10 (d). The visualized surface is very different from the original shape and is much worse than the surface reconstructed with our proposed method. In fact, the ICP results for all conditions were even worse than the surface estimated from a single range image.

Figure 10 shows the reconstructed surface and the initial input point cloud. Figure 10 (a) and (b) show the initial point cloud and the point cloud after the method was applied. Thirty range images are illustrated in different colors. Figure 10 (c) shows the reconstructed surface using the base frame with the 3,525-point dataset. Figure 10 (d) shows the surface reconstructed by ICP. It can be seen that ICP was not suitable for a sparse point cloud. Figure 10 (e) and (f) show the surface reconstructed by our method using the 3,525-point dataset. Noise was added to
the input data of Fig. 10 (f). Figure 10 (g) shows the surface reconstructed using the 6,981-point dataset. Figure 10 (h) shows the surface of a model whose number of vertices was reduced to the same level as Fig. 10 (g) from the original surface shown in Fig. 7. Comparing these two surfaces, our method reconstructed the surface accurately using a sparse point dataset.

In addition, we evaluated the newly devised techniques described in Section 3.1. In the experiment, we used the same dataset as that described above, containing 3,525 points. We compared the proposed method and four other variations. Figure 11 shows the results. The horizontal axis represents the number of cycles, and the vertical axis represents the error, calculated as the average of $B$. The light blue line shows the result obtained by using the original proposed method. The purple line shows the result without feature (II) described in Section 3.1 (the exponential term in equation (6) is removed). The red line shows the result without feature (IV) (the weight function is removed from equation (18)). The blue line shows the result without both (II) and (IV). The green line shows the result without (II), (III), (IV), and (V).

As described in Section 3.1, we estimated the surface after the last cycle with the decreased smoothing parameter $\varepsilon_0$. Therefore, the error at the right end of each line was drastically reduced.

As shown in Fig. 11, feature (IV) was more effective for improving the accuracy than feature (II). Also, the result without features (II) and (IV) was worse. In addition, the result without features (II)–(V) was better than the result without features (II) and (IV). This is because iterative normal estimation using a low-accuracy surface caused large errors compared with the normals estimated initially.

We also compared the normal maps of the reconstructed shapes with that of the object model at five different resolutions. The accuracy was evaluated by calculating the average angle between true normals and estimated normals. Figure 12 shows the result. For all resolutions, the accuracy of the normals was improved.

4.2 Reconstruction Using Complicated Model

In this section, we present the results obtained when using a more complex shape. In this experiment, we used Lucy [16] as the model. Our results showed that the proposed method is useful for the case where the surface reconstructed using a single frame has fewer shape features than the original object. Figure 13 (a) shows a rendering of the original image of the object. The number of mesh triangles that make up the head of Lucy was 378,907, and the size of the object was $88 \times 86 \times 110$ (original units: mm). The movement per frame was set by setting the axis of rotation to $l = (0, 1, 1)$, the angle of rotation to $\theta = 0.03$ rad, and the translation to $v = (0.1, -0.1, 0.2)$. The total number of frames was 40. The total number of points used for the estimation was 4,028 (about 100 per frame), which was 2% of the original number of points. Other parameters were the same as in the previous section.

Figure 13 (c) shows the surface of a model whose number of vertices was reduced to the same level of input data whose number of points was 4,000, Fig. 13 (b) shows the surface reconstructed using a single frame, and Fig. 13 (d) shows the result. In the surface reconstructed by the proposed method, even in the case where the total number of input points was 2% of the original, there were many features that could not be seen in the surface reconstructed using only a single frame.

4.3 Reconstruction by Scanning Actual Data

In this section, we applied the method to actual data obtained by a structured-light sensing system to show how it performed.

![Figure 11](image_url) Average and variance of $B$ for range images with several variations of our method.

![Figure 12](image_url) Average angle between true normals and estimated normals at several resolutions.

![Figure 13](image_url) Reconstructed surface of complicated model. (a) Original surface of the model. (b) Surface reconstructed using a single frame. (c) Surface of model whose number of vertices was reduced to 4,000. (d) Surface reconstructed by proposed method.
Figure 14 (a) shows the sensor system we used. Two cameras captured the scene at a frame rate of 500 fps. The resolution was $1,280 \times 1,024$. A 15-line pattern was projected as structured light. Each camera obtained 3D points by observing the projected pattern, and the two sets of points were aligned to a unique coordinate system. The image captured during the measurement is also shown in Fig. 14 (a). The measured target was a plaster figure of Venus (Fig. 14 (b)). The size of the target was $136 \text{mm} \times 171 \text{mm} \times 133 \text{mm}$. The total number of frames was 30, and we set the 15th frame as the base frame. The total number of 3D points was 16,966.

In this experiment, we slid the target as shown in Fig. 14 (a). As mentioned in Section 1, we also believe that there are a lot of attractive applications in which the motion tends to be linear and the direction and the speed are unknown. The settings in Section 4.3 are reasonable for verifying the feasibility of the method for those applications. Furthermore, we plan to use range images that are acquired in a short time using high-speed imaging. In this case, the target motion can be assumed to be uniform.

Figure 14 (d) shows the surface reconstructed using a single frame, and Fig. 14 (f) shows the surface reconstructed by the proposed method. Figure 14 (c) and (e) show the initial point cloud and the point cloud after the method was applied. In the surface reconstructed by the proposed method, we can see features of the nose, mouse, and hair that cannot be seen in the surface reconstructed using only a single frame. We also confirmed that similar linear motion could be estimated despite having no advance information about the motion.

5. Conclusion

In this paper, we have proposed a method of reconstructing a high-resolution surface using multiple range images which are obtained by observing an unknown moving target. The proposed method consists of three processes: shape estimation, motion estimation, and normal estimation. The proposed method can control the local weight for estimation, which contributed to improved accuracy and made the method more robust for sparse point clouds.

In the experiments, we applied the method to range images with different resolutions and levels of noise, other complex shapes, and to actual data. The method worked well, and shape features that could not be obtained by using a single range image appeared by integrating multiple range images. In addition, it could be applied to very low-resolution range images. Furthermore, our method achieved good accuracy in the experiments described in Sections 4.1 and 4.2, whose input data involved large rotational and translational motions. Our method also achieved good reconstruction for noisy input data whose noise was larger than that in common measurement conditions. Therefore, we fully confirmed that this method has no problem in general environments.

In future work, it will be important to adjust the smoothing parameter in the shape estimation. In addition, in order to increase the accuracy of motion estimation, improvement of the initial shape estimation will also be needed for giving more stable initial values. It is believed that real-time sensing may also be possible...
by adopting linearization or approximation in the nonlinear optimization to speed-up the calculation.

References


© 2013 Information Processing Society of Japan

Shohei Noguchi was born in 1988. He received his B.E. degree in Mathematical Engineering and Information Physics from the University of Tokyo in 2012. Since then, he has been a master course student of the Graduate School of Information Science and Technology, the University of Tokyo. His research interests include computer vision, high-speed 3D sensing and digital archiving.

Yoshihiro Watanabe was born in 1979. He received his B.E. degree in Mathematical Engineering and Information Physics from the University of Tokyo in 2002. He received his M.E. and Ph.D. degrees in Information Science and Technology from the University of Tokyo in 2004 and 2007, respectively. He became a research fellow at the University of Tokyo in 2007, a project assistant professor at the University of Tokyo in 2008, and an assistant professor at the University of Tokyo in 2011. His research interest is high-speed image processing and its applications including robotics, inspection, video media engineering, man-machine interfaces and digital archiving. He is a member of RSJ and VRSJ.

Masatoshi Ishikawa received his B.E., M.E., and Dr. Eng. degrees in Mathematical Engineering and Information Physics in 1977, 1979, and 1988, respectively, from the University of Tokyo, Tokyo, Japan. He was a senior researcher in the Industrial Products Research Institute, Tsukuba, Japan, from 1979 to 1989. From 1989 to 1999, he was an associate professor at the University of Tokyo. From 1999, he was a professor at the University of Tokyo. From 2004 to 2005 and from 2005 to 2006, he was a vice-president and an executive vice-president at the University of Tokyo, respectively. His current research interests include parallel computing, smart sensors, vision chips, sensor fusion, and robotics.

(Communicated by Ryusuke Sagawa)