

A Range Image Refinement Technique for Multi-view 3D Model Reconstruction

Soon-Yong Park and Murali Subbarao
Electrical and Computer Engineering
State University of New York at Stony Brook, USA
E-mail: parksy@ece.sunysb.edu

Abstract

This paper presents a range image refinement technique for generating accurate 3D computer models of real objects. Range images obtained from a stereo-vision system typically experience geometric distortions on reconstructed 3D surfaces due to the inherent stereo matching problems such as occlusions or mismatchings. This paper introduces a range image refinement technique to correct such erroneous ranges by employing epipolar geometry of a multi-view modeling system and the visual hull of an object. After registering multiple range images into a common coordinate system, we first determine if a 3D point in a range image is erroneous, by measuring registration of the point with its correspondences in other range images. The correspondences are determined on 3D contours which are inverse-projections of epipolar lines in other 2D silhouette images. Then the range of the point is refined onto the object's surface, if it is erroneous. We employ two techniques to search the correspondences fast. In case that there is no correspondence for an erroneous point, we refine the point onto the visual hull of the object. We show that refined range images yield better geometric structures in reconstructed 3D models.

1. Introduction

Generating 3D computer models of real objects is of much interest in *Computer Vision* and *Computer Graphics*. One of the recent research interests is generating a complete and closed 3D model by merging multiple images of an object. One common technique is employing a single or multiple ranging sensors to acquire and merge multi-view range images [1, 4]. There is a variety of ranging techniques to obtain range images of a scene of interest. In order to generate accurate 3D models, most investigations employ active ranging techniques such as laser ranging, structured

light pattern, space-time coded pattern, etc.

In contrast, passive techniques work only on naturally formed images produced by reflected light from the object. A common technique is Stereo Image Analysis (SIA). However, due to inherent stereo problems, relatively fewer number of researchers have employed stereoscopic imaging sensors to generate complete 3D models [3, 5, 14]. Range images obtained from a stereo ranging sensor typically experience erroneous points on reconstructed 3D partial surfaces. In order to obtain accurate range images using SIA, some researchers project random dot patterns onto object's surfaces to enhance contrast on them [9]. Others employ multi-baseline or multi-resolution [14] technique to reduce the number of stereo mismatchings. Even using these techniques, obtaining accurate range images is still difficult in some portions of the object's surfaces.

One simple approach of removing erroneous points is employing a linear low-pass filter or a non-linear filter [4, 5, 12]. However, it is not easy for these approaches to remove some erroneous regions where the errors are dominant within a filtering mask. Another simple approach is obtaining a large number of range images to average out the errors [4, 10, 13]. However, acquiring multiple range images using SIA is a computationally expensive task. Another approach is employing a range image registration technique, such as Iterative Closest Point (ICP), to reduce noises in range images [11]. Some researchers try to remove these errors using the visual hull of the object [7, 8]. However, this technique can remove only those outside the visual hull but not inside.

In this paper, we present a range image refinement technique to reconstruct accurate 3D models of real objects. After registration of multi-view range images, we first determine if a 3D point in a range image is erroneous, by measuring the registration of the point with its conjugates in other range images. The conjugate points are searched on 3D contours which are inverse-projections of epipolar lines in other 2D silhouette images. When the 3D point is decided as erroneous or it is outside the visual hull of the object, we refine the range of the point to the mean of the conjugates or

onto the surface of the visual hull. In order to make searching and refinement fast, we employ two techniques, point-to-projection search and ordering constraint. Experimental results on a ground truth object and real objects show that 3D models from refined range images yield better geometric structures.

2. Range Error due to Stereo Mismatching

Range images obtained by a stereo camera could experience some erroneous points on their surfaces due to stereo mismatching. An example of erroneous range surface is shown in Figure 1. Figure 1(a) shows a picture of an object and Figure 1(b) shows a reconstructed mesh model of the front partial surface of the object, which is seen from the top. As in the figure, stereo mismatching errors appear as sharp peaks on the partial shape.

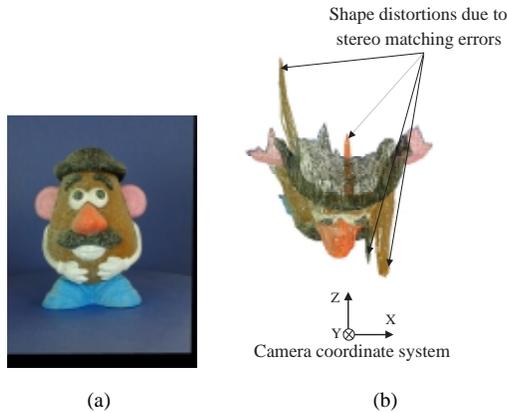


Figure 1. An example of stereo mismatching (a) An object (b) A partial shape of the object shows some erroneous surfaces

When we compute the depth of a 3D point \mathbf{P} , which maps to an image point \mathbf{p} , the 3D point is on the ray of \mathbf{p} . Suppose an object is placed considerably far from the camera, then the ray is almost parallel to the \mathbf{Z}_c axis (viewing direction) of a camera coordinate system. Therefore, we can estimate that the coordinates of mismatching errors, between a correct points set $\{\mathbf{P}\}$ and an erroneous points set $\{\mathbf{P}'\}$, distribute mostly on the \mathbf{Z}_c axis. An example of error distribution between two partial surfaces in Figure 2(a) is plotted in Figure 2(b). In Figure 2(a), the top range image has some errors, and the bottom one has negligible errors in contrast. As shown in Figure 2(b), most errors are distributed along the \mathbf{Z}_c axis, which coincides with the viewing vector of the range image.

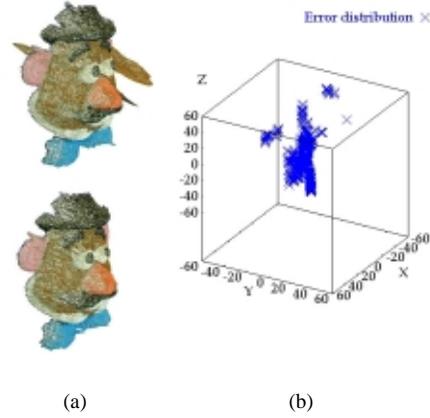


Figure 2. Error distribution in 3D space. (a) Top: A range image with erroneous surfaces. Bottom: The same image without error. (b) Most errors are distributed along the \mathbf{Z}_c axis of the camera coordinate system

3. Refinement Methodology

3.1. Erroneous 3D points

As presented in the previous section, stereo mismatching errors in a range image occur mostly on such object's surfaces whose normal vectors are in a high angle with respect to the viewing direction of the image. However, they may be seen better from another viewing direction because the object is scanned from multiple directions. Therefore, it is possible to refine an erroneous 3D point, if we know its correspondences on other view's range images.

Suppose there is a 3D point \mathbf{P} on an object's surface and two different views \mathcal{V}_i and \mathcal{V}_j can see the point. Then we can write two representations of the point as \mathbf{P}_i^i and \mathbf{P}_j^i in the corresponding range images \mathcal{R}_i and \mathcal{R}_j , based on the view point of \mathcal{V}_i . When the two points are registered to a common coordinate system, their coordinates should coincide in an ideal case. However, in a real case, they may not coincide because of systematic or calibration error of the vision system.

If the two points do not coincide in 3D space, a problem arises that how we can determine the correspondences between multiple range images. We assume that all correspondences occur on the ray of a 3D point. Considering \mathcal{R}_i as a reference range image and \mathbf{P}_i^i as a reference point, we find all intersections in the other range images on the ray of the reference point.

Suppose there are two range images \mathcal{R}_i and \mathcal{R}_j as shown

in Figure 3(a) and they are registered to a common view \mathcal{V}_i . Assuming registration of the range images is very accurate, we know that \mathbf{P}_i^i is very close to another point \mathbf{P}_j^i , which is on the line of sight of the point \mathbf{P}_i^i from \mathcal{V}_i . If the origin of the view \mathcal{V}_i is \mathbf{O}_i , the correspondence of \mathbf{P}_i^i on \mathcal{R}_j is on the ray $\mathbf{L} = \alpha \mathbf{P}_i^i, \alpha \in \mathbb{R}$. Therefore, we write

$$\mathbf{P}_j^i = \mathcal{R}_j \cap \mathbf{L} \quad (1)$$

$$\text{and } \mathbf{P}_j^i \times \mathbf{L} = 0. \quad (2)$$

Therefore, the two matching points \mathbf{P}_i^i and \mathbf{P}_j^i in Figure 3(a) are very close and there is only a small registration error ϵ between them such that

$$\epsilon = \|\mathbf{P}_i^i - \mathbf{P}_j^i\| \approx 0. \quad (3)$$

Let us consider another 3D point \mathbf{P}_i^i as shown in Figure 3(b). This point is considered as an erroneous point in the range image \mathcal{R}_i due to stereo mismatching, while its conjugate \mathbf{P}_j^i on the \mathcal{R}_j is a correct surface point. Then there should be significant registration error between two points as in the figure, which is larger than a threshold d_T such that

$$\epsilon = \|\mathbf{P}_i^i - \mathbf{P}_j^i\| > d_T. \quad (4)$$

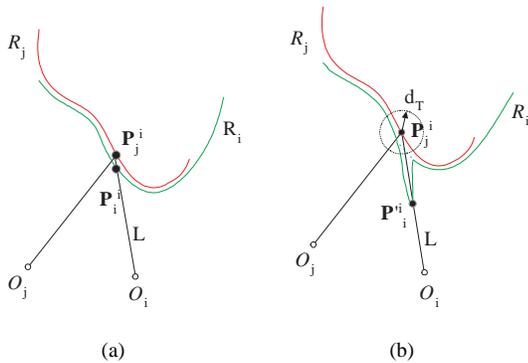


Figure 3. A typical error on a partial shape due to stereo mismatching. (a) Two matching 3D points are very close (b) \mathbf{P}_i^i is far from \mathbf{P}_j^i

If multiple range images are intersected by the ray \mathbf{L} , it is possible to determine several correspondences in the range images. Suppose there are K range images \mathcal{R}_j for $j = 0, \dots, K - 1$, which are intersected by \mathbf{L} . Then we determine if the point \mathbf{P}_i^i is

$$\text{erroneous} \begin{cases} \text{true, if } \|\mathbf{P}_i^i - \mathbf{P}_k^i\| > d_T, \forall k, 0 \leq k \leq K - 1 \\ \text{false, if } \|\mathbf{P}_i^i - \mathbf{P}_k^i\| < d_T, \exists k, 0 \leq k \leq K - 1 \end{cases} \quad (5)$$

In addition to the error measure, we also use the visual hull of an object to decide if a 3D point is erroneous. When the 3D point is outside the visual hull of the object, we consider it is erroneous. This technique is the same as what the shape-from-silhouettes technique does. In summary, let $VH(O)$ be the visual hull of an object O , then the 3D point \mathbf{P}_i^i is also

$$\text{erroneous} \begin{cases} \text{true, if } \mathbf{P}_i^i \notin VH(O) \\ \text{not determined, if } \mathbf{P}_i^i \in VH(O) \end{cases} \quad (6)$$

We first check if the point is outside the visual hull. If the point is outside, we consider it is erroneous and refine it. In case that the point is inside the visual hull, we use Equation (5) to check if it is erroneous.

3.2. Correspondence on Epipolar Contour

In order to refine a reference 3D point in a reference range image, we search all its correspondences in the other range images. Suppose again there is a reference 3D point \mathbf{P}_i (or \mathbf{P}_i^i) on \mathcal{R}_i as shown in Figure 4. The perspective projection of the 3D point to the image plane of \mathcal{V}_i is a 2D image point \mathbf{p}_i . Let us consider a problem to find the correspondences of \mathbf{P}_i in other range images, \mathcal{R}_j and \mathcal{R}_k in this example. We employ the epipolar geometry of our system. If we know the fundamental matrix between \mathcal{V}_i and \mathcal{V}_j , we can restrict searching of intersections on the epipolar line \mathbf{u}_j which is

$$\mathbf{u}_j = \mathbf{F}_{ij} \mathbf{p}_i, \quad (7)$$

where \mathbf{F}_{ij} is the fundamental matrix between \mathcal{V}_i and \mathcal{V}_j .

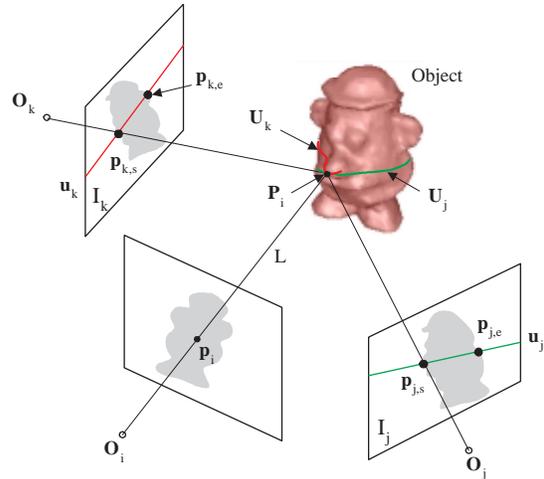


Figure 4. Epipolar geometry of a multi-view system

On the image plane I_j in the figure, the epipolar line \mathbf{u}_j overlaps with a silhouette of the object, which is a binary

representation of the perspective projection of the object to \mathcal{V}_j . On the object's silhouette, we determine a starting point $\mathbf{p}_{j,s}$ and an ending point $\mathbf{p}_{j,e}$ on the epipolar line. From the starting point to the ending point, we then inverse project all 2D image points on the epipolar line \mathbf{u}_j to corresponding 3D points from $\mathbf{P}_{j,s}$ to $\mathbf{P}_{j,e}$ on the surface of \mathcal{R}_j . All the 3D points on the surface then form a 3D contour $\{\mathbf{U}_j\}$

Along the 3D contour $\{\mathbf{U}_j\}$, we search all intersections by the ray \mathbf{L} , which satisfy Equation (1) and (2). Then from these candidates we pick one of them as $\mathbf{P}_{j,c}$, the correspondence of the point \mathbf{P}_i . The intersection point $\mathbf{P}_{j,c}$ satisfies following equations.

$$\{\mathbf{U}_j\} = \{\mathbf{P} | \mathbf{P} \in \mathbf{T}_j^i \mathbf{M}_j^{-1} \{\overline{\mathbf{p}_{j,s} \mathbf{p}_{j,e}}\}\} \quad (8)$$

$$\mathbf{P}_{j,c} \times \mathbf{P}_i = 0 \text{ and } \mathbf{P}_{j,c} \in \{\mathbf{U}_j\}. \quad (9)$$

, where \mathbf{M}_j is the perspective transform matrix of the j th view, and \mathbf{T}_j^i is the transformation of coordinate system from the j th view to the i th view. Because there could be several intersections on the 3D contour by the ray, we find all of them by checking signs of cross products of the vector of \mathbf{L} and other vectors to points on $\{\mathbf{U}_j\}$. More details are in the next section.

4. Range Image Refinement

4.1. Correspondence Search

In order to search all intersections on the 3D contour $\overline{\mathbf{P}_{j,s} \mathbf{P}_{j,e}}$, we first find all zero crossings of the cross product $\mathbf{P}_j \times \mathbf{P}_i$, where $\mathbf{P}_j \in \overline{\mathbf{P}_{j,s} \mathbf{P}_{j,e}}$. Because the Y_s axis of the turntable's coordinate system is almost parallel to the Y_c axis of the camera coordinate system, we check sign of the Y component of the product to search zero crossings. If the sign changes, we consider there happens a zero crossing.

Figure 5 shows an example of zero crossings on a 3D contour. Of course there could be multiple (N_c) intersections on the contour, for example in the figure from $\mathbf{P}_{j,c,0}$ to \mathbf{P}_{j,c,N_c-1} . However, from the point of i th view point, only one intersection is visible, which is the closest to \mathbf{P}_i . In this figure therefore, we take $\mathbf{P}_{j,c,0}$ as the correspondence $\mathbf{P}_{j,c}$ on j th view's contour $\{\mathbf{U}_j\}$. To avoid false correspondence, we test and remove some intersections

- whose normal vectors are in a high angle with respect to their original view points,
- which are hidden from the reference view point,
- or which are outside the visual hull of the object.

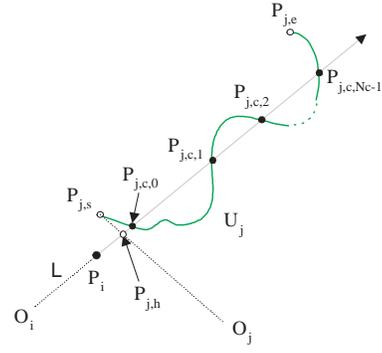


Figure 5. Intersections on a 3D contour by a reference vector

4.2. Fast Searching Technique

In this section, we consider computation time of intersection searching and introduce fast searching techniques. Suppose there are Q points on a 3D contour and computation for intersection searching takes $O(Q)$ operations. If there are K overlapping ranges from a reference point, we need to search intersections on K contours, thus $O(KQ)$ computations. In order to refine a range image, whose object area is equivalent to $M_X \times M_Y$ image size, then we need to compute $O(KQM_X M_Y)$ operations. To refine all N range images, we finally need $O(NKQM_X M_Y)$ operations. These computations are computationally very expensive. In a typical Pentium-4 1.8GHz computer, it takes more than 30 minutes to refine 16 range images with 320×240 resolution. In order to make the computation time fast, we employ two techniques based on point-to-projection range searching and ordering constraint.

4.2.1 Point-to-Projection Searching

Instead of searching intersections on a 3D contour, we use a point-to-projection technique to solve the correspondence problem as shown in Figure 6. To find the j th view's correspondence from a reference point \mathbf{P}_i , we project it to the image plane of \mathcal{V}_j to a 2D point \mathbf{p}_j^i , inversely project the 2D point to j th range image \mathcal{R}_j to get \mathbf{P}_j^i . In other words,

$$\mathbf{P}_j^i = \mathbf{T}_j^i R_j(\mathbf{M}_j \mathbf{T}_i^j \mathbf{P}_i), \quad (10)$$

where $R_j(\mathbf{p})$ is the j th view's range of a 2D point \mathbf{p} .

By searching all inverse-projection points \mathbf{P}_k^i , $k = 0 \dots K - 1$ in other range images, we use Equation (5) to determine if the reference point is erroneous or not. Inverse-projection is not exactly an intersection point by the ray. However, if two range surfaces are very closely registered, we can approximate the point as an intersection. Once the

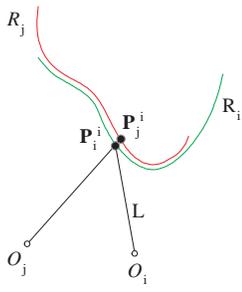


Figure 6. Point-to-Projection searching

reference point is determined as erroneous, we then find the real intersections on the ray of L using the technique presented in Section 3.2. Using this searching technique, we refine all range images in about 5 or 10 minutes depending on objects.

4.2.2 Ordering Constraint

The next fast searching technique is using ordering constraint in a pair of multi-view images. Once a reference point P_i^i is determined as erroneous, we find its conjugates on a 3D contour $\{U_j\}$, which is the inverse projection of a 2D segment on u_j . In Section 3.2, we search the conjugates along u_j starting from $p_{j,s}$ to $p_{j,e}$. However, instead of all points on the epipolar line we use the ordering constraint between a pair of multi-view images. Let us consider an example in Figure 7. As in the figure, suppose a 2D reference pixel $p_i(x_i, y_i)$ on I_i matches with its conjugate $p_j(x_j, y_j)$ on I_j , where (x_i, y_i) and (x_j, y_j) are their coordinates on I_i and I_j respectively. In other words, their 3D conjugates P_i^i and P_j^i match each other.

Now consider another reference point $p_i(x_i + 1, y_i)$ on I_i , which is the next pixel of $p_i(x_i, y_i)$. Then we know that its matching pixel on I_j is always on the right-side of $p_j(x_j, y_j)$ according to the ordering constraint. Therefore we search an intersection of the ray L on a 3D contour along $\{U_j\}$, which is the inverse projection of a line segment $\overline{p'_{j,s}p'_{j,e}}$ rather than $\overline{p_{j,s}p_{j,e}}$. Based on the constraint, the x coordinate of the starting point could be the same with that of $p_j(x_j, y_j)$, that is x_j . However, in order to find a zero crossing on the contour, we give a margin in the left direction. As a result, we set the range of intersection searching along the x axis from $x_j - \delta_-$ to $x_j + \delta_+$. By assuming there is no sudden depth change on the object's surface, we set δ_- to 2 and δ_+ to 10 pixels.

4.3. Refinement onto Visual Hull

Some erroneous range points are outside the visual hull ($VH(O)$) of an object. If a 3D point is outside of $VH(O)$,

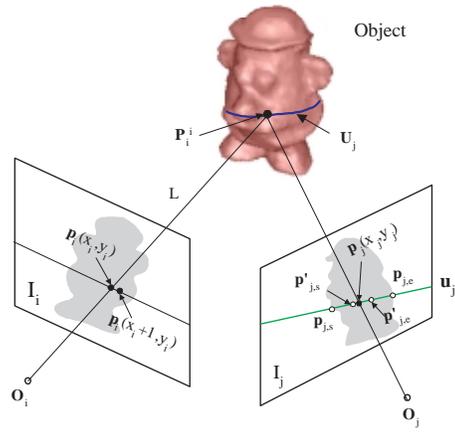


Figure 7. Reducing search range using ordering constraint

we have to refine it to a correct 3D point on the object's surface. If there are enough matching points in other range images, we can refine it using the presented techniques. However, a problem arises if there is not enough or no matching point. In this case, we decide to refine it onto the visual hull of the object.

As shown in Figure 5, a visual hull point $P_{j,h}$ can be found by the intersection of two rays, L and $\overline{P_{j,s}O_j}$. Intersection of two non-intersecting rays can be solved using a simple linear equation. Because there are multiple views, we acquire multiple visual hull points $P_{k,h}$, $0 \leq k \leq K-1$. Then the problem is now deciding which one is on $VH(O)$ of the object. Let us consider an example in Figure 8. On the ray L from a view origin O_i , we acquire multiple visual hull points from $P_{0,h}$ to $P_{3,h}$. To determine which one is on $VH(O)$, we simply project every point to image planes of all multiple views and check if it is on $VH(O)$. In this figure, $P_{3,h}$ is on the visual hull. Then we move P_i to $P_{3,h}$.

On a line segment $\overline{p_{j,s}p_{j,e}}$ on an epipolar line u_j in Figure 4, we get two visual hull points using two rays $\overline{P_{j,s}O_j}$ and $\overline{P_{j,e}O_j}$. However, we take only one of them from $\overline{P_{j,s}O_j}$, because the point P_i can see only the starting point $P_{j,s}$. After acquiring all visual hull points from K overlapping views, we sort them according to distance from the reference point and check if they are inside $VH(O)$.

4.4. Refinement Steps

Range image refinement steps are as follows. For every 3D point in a range image, we first determine if it is erroneous or not using two methods. One is to determine if it is outside of the visual hull and the other is to determine if it is erroneous using the point-to-projection searching technique presented in Section 4.2.1. If the point is erroneous, we

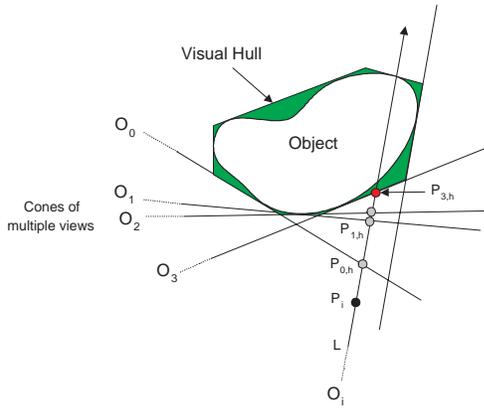


Figure 8. Refinement of P_i onto a visual hull point $P_{3,h}$

search all intersections on surfaces of other range images. If there are at least two intersections $P_{j,c}$, $0 \leq j < K - 1$, we change the coordinates of the point to the mean P_m of the all intersections if $\sigma_p^2 < \sigma_T^2$. Here P_m and σ_p^2 are mean and variance of the points $\sum_j P_{j,c}^i$ and σ_T^2 is a threshold for the variance.

When there are not enough intersections or $\sigma_p^2 > \sigma_T^2$, but if the reference point is outside visual hull, we change the coordinates of P_i to $P_{j,h}$ as presented in Section 4.3. In this case $P_{j,h}$ should be also inside the visual hull of the object.

5. Experimental Results

5.1. Range Image Acquisition

To generate multi-view range images, we use a stereo-based range image acquisition system, which consists of a stereo camera, a turntable stage, and a personal computer. A range image is obtained from a pair of stereo images. In order to find correspondence in the image pair, we use a multi-resolution stereo matching technique [2, 8]. The stereo pair is resampled to form a Gaussian image pyramid and a simple SSD-based (Sum of Squared Difference) correlation technique is performed at each pyramid level. Multi-view range images are obtained by rotating the turntable with a fixed rotation angle according to number of views. We acquire range images of an object from 8 or 16 ($N = 8, 16$) different view directions. We use a Pentium-4 1.8GHz personal computer for our vision system. A range image resolution for each view is 320×240 in row and column directions. For every range image \mathcal{R}_i , we refine it pixel by pixel (or point-by-point). In order to refine an erroneous point, we search its correspondences in range images from $\mathcal{R}_{-\frac{N}{4}}$

to $\mathcal{R}_{+\frac{N}{4}}$ ($K = N/2$). We set the error threshold d_T to 3 mm and the variance threshold σ_T^2 to 2 mm.

5.2. Error Analysis

We generate 3D models of a ground truth object to analyze and compare reconstruction errors of before and after refinement. Figure 9(a) shows a picture of a cylindrical object for error analysis. Figure 9(b) is the dimension of the object. We acquire range images of the object from 8 or 16 view points and register and integrate them to a 3D mesh model. Then we measure dimensional and registration errors between the 3D model and a ground truth model. To generate the ground truth model, we reconstruct a 3D model of dense point clouds. We synthesize the object's surface for every 1 mm² to produce about 50000 points to represent its outer surface.

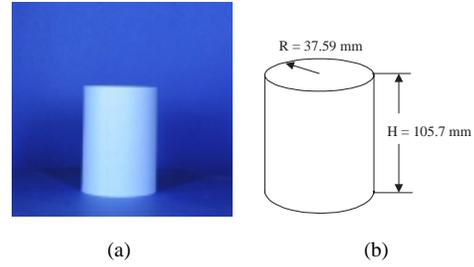


Figure 9. 'Cylinder' object for error analysis (a) Picture of the object (b) Dimension of the object

The two 3D models, the reconstructed model and the ground truth model, are registered into a common coordinate system using ICP registration technique. After registration, we measure mean and variance of the radius of the reconstructed model. Table 1 shows results of mean and variance of the radius of the object. We compare errors of two 3D models which are generated by merging original range images and refined range images. We show the errors when the models are merged from 8 views and 16 views. To show the results with different size of voxel, we reconstruct different 3D models by changing voxel size of 2 mm, 2.5 mm, and 3 mm. We also measure RMS and maximum errors of closest distance between a 3D point on the reconstructed model and the ground truth model. Table 2 shows results of these errors. As shown in both tables, 3D models generated from refined range images have better geometric structures than those from originals.

Table 1. Mean (m) and variance (v) error of the radius of 'Cylinder' object in mm

N	voxel	Original Range			Refined Range		
		2.0	2.5	3.0	2.0	2.5	3.0
8	m	36.32	36.32	36.33	36.35	36.35	36.36
	v	0.416	0.438	0.40	0.316	0.307	0.313
16	m	36.40	36.39	36.41	36.41	36.41	36.41
	v	0.280	0.296	0.317	0.232	0.253	0.252

Table 2. RMS and MAX error of the radius of 'Cylinder' object in mm

N	voxel	Original Range			Refined Range		
		2.0	2.5	3.0	2.0	2.5	3.0
8	rms	1.33	1.33	1.32	1.27	1.27	1.26
	max	3.56	3.60	3.37	3.58	3.14	3.32
16	rms	1.21	1.22	1.21	1.19	1.20	1.20
	max	2.63	2.54	2.60	2.50	2.25	2.79

5.3. Real Objects

The proposed technique is also tested on real objects. Figure 10 shows an example of a single line refinement of 'Potatohead' object shown in Figure 1(a). Figure 10(a) shows the original front view (\mathcal{V}_0) of the object and a 3D contour which corresponds to a single horizontal line on its 2D plane. In Figure 10(b), we plot all corresponding profiles from $\mathcal{V}_{N-\frac{N}{4}}$ to $\mathcal{V}_{+\frac{N}{4}}$ on the XZ plane of the camera coordinates. On the plot of \mathcal{V}_0 , there are some erroneous points as shown the contour in Figure 10(a). A refined range image and all profiles for the same horizontal scan lines are shown in Figure 10(b) and (d).

Some of original and refined range images of the object are shown in Figure 11(b) and (c). Figure 11(b) shows there are some artifacts in range images due to stereo mismatching. Some bright and dark portions on the object area represent shape distortions on surface. The refined range images in Figure 11(c) show that their surface structures are refined compared to their originals. Processing time of range refinement and number of range pixels on all 16 images are shown in Table 3.

Table 3. Processing time of refining range images

Object	Potatohead	Soccerball
Number of vertices	264608	156662
Refinement time (sec)	510.0	122.0

Comparison of two 3D models is shown in Figure 12(a) and (b). In this figure, the size of a voxel grid is 2.5 mm . The shaded surface models show the back surface of the object. The refined 3D model in Figure 12(b) shows better surface structure than the original in Figure 12(a). Figure 13 shows reconstruction results of another object 'Soccerball'. Figure 13(b) is the 3D model from original range images, and Figure 13(c) is from refined images. The figure also shows that refined range images yield better surface structure.

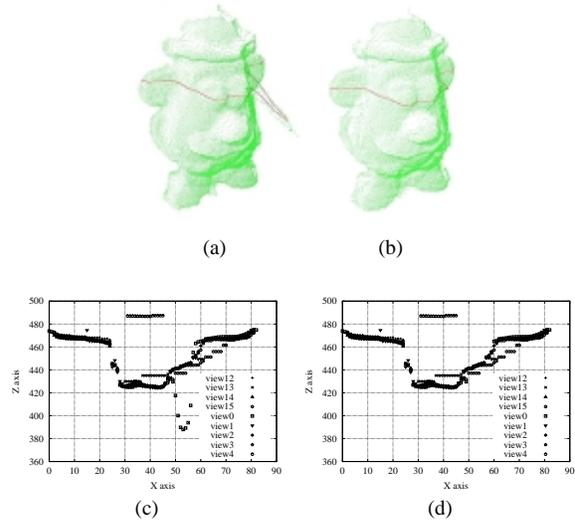
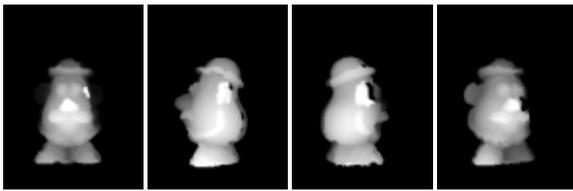


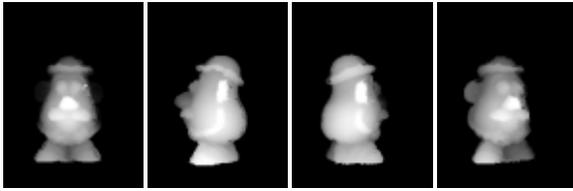
Figure 10. An example of range refinement. (a) A horizontal scan line is plotted on a original partial shape. (b) The same line is plotted on a refined shape (c) Profiles of all overlapping range contours before refinement (d) Profiles of all overlapping range contours after refinement

6. Conclusions

We present a range image refinement technique to generate accurate 3D computer models. The proposed technique employs the epipolar geometry of our vision system and the visual hull of an object to refine erroneous 3D points. After multi-view range images are registered to a common coordinate system, we first determine if a reference 3D point in a range image is erroneous by measuring registration error of the point to its correspondences in the other overlapping range images. The correspondences are intersections by the ray of the reference point. Then the range of an erroneous point is refined onto the object's surface. Visual hull of the object is also employed to decide if the point is erroneous

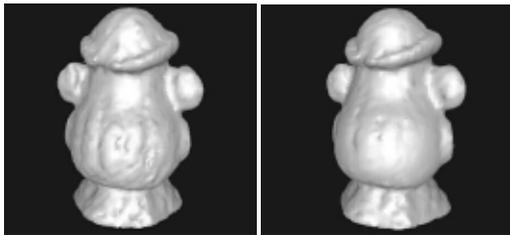


(a)



(b)

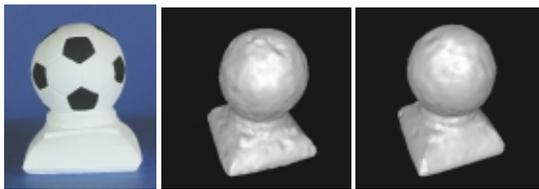
Figure 11. Refinement results of 'Potatohead' object. (a) Some original range images (b) Refined range images



(a)

(b)

Figure 12. Reconstruction results of 'Potatohead' object. (a) A shaded surface on the original 3D model (b) The same surface on the refined 3D model



(a)

(b)

(c)

Figure 13. Reconstruction results of 'Soccerball' object. (a) Picture of the object (b) A shaded surface on the original 3D model (c) The same surface on the refined 3D model

and to refine it onto the visual hull. Experimental results show the proposed technique yields better geometric structures on reconstructed 3D computer models.

References

- [1] F. Bernardini, I. M. Martin, and H. Rushmeier, "High-quality Texture Reconstruction from Multiple Scans," *IEEE Transactions on Visualization and Computer Graphics*, 7(4):318-332, 2001.
- [2] P.J. Burt, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, 31(4):532-540, 1983.
- [3] Q. Chen and G. Medioni, "A Volumetric Stereo Matching Method: Application to Image-Based Modeling," *Proceedings of Computer Vision and Pattern Recognition*, 29-34, 1999.
- [4] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *Proceedings of SIGGRAPH*, 303-312, 1996.
- [5] R. Koch, "Surface Segmentation and Modeling of 3-D Polygonal Objects from Stereoscopic Image Pairs," *Proceedings of International Conference on Pattern Recognition*, Vienna, 1996.
- [6] W. Matusik, C. Buehler, R. Raskar, L. McMillan, and S. J. Gortler, "Image-Based Visual Hulls", *SIGGRAPH*, 2000.
- [7] W. Niem and R. Buschmann, "Automatic Modelling of 3D Natural Objects from Multiple Views," *In European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production*, 23-24, 1994.
- [8] S. Y. Park and M. Subbaro, "Automatic 3D Model Reconstruction using Voxel Coding and Pose Integration," *Proceedings of International Conference on Image Processing*, 533-536, 2002.
- [9] S. Radim, "Accurate Natural Surface Reconstruction from Polynocular Stereo," *Proceedings of NATO Advanced Research Workshop*, 69-86, 2002.
- [10] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-Time 3D Model Acquisition," *Proceedings of SIGGRAPH*, 2002.
- [11] R. Sagawa, T. Oishi, A. Nakazawa, R. Kurazume, and K. Ikeuchi, "Iterative Refinement of Range Images with Anisotropic Error Distribution," *Proceedings of 2002 IEEE International Conference on Intelligent Robots and Systems*, 79-85, 2002.
- [12] Y. Sheng and B. Chen, "Measuring of a Three-Dimensional Surface by Use of a Spatial Distance Computation," *Applied Optics*, 42(11):1958-1972, 2003.
- [13] M. D. Wheeler, Y. Sato, and K. Ikenuchi, "Consensus Surfaces for Modeling 3D Objects from Multiple Range Images", *IEEE Conference on Computer Vision*, 917-924, 1998.
- [14] R. Yang, G. Welch, and G. Bishop, "Real-Time Consensus-Based Scene Reconstruction using Commodity Graphics Hardware," *Proceedings of Pacific Graphics*, 2002.