3-D Reconstruction using Surface Detection

Woo-Seok Jang, Kwan-Jung Oh, and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju, Republic of Korea

ABSTRACT

In this paper, we propose a three-dimensional (3-D) reconstruction algorithm using surface detection. The proposed method consists of three parts: object detection, surface detection, and texture mapping. In object detection, we examine whether each voxel in the 3-D space is a part of the object or not. We separate only the surface from the detected object and then determine the texture value of the surface from the multiple 2-D images. Experimental results demonstrate the feasibility and accuracy of our approach.

1. INTRODUCTION

The three-dimensional (3-D) object reconstruction from multiple two-dimensional (2-D) images is a very active research field. The aim of this research is to reconstruct geometry and color information of the 3-D object from a given sparse set of images [1]. If the multiple 2-D images are not calibrated, the correspondences between given images should be determined and the quality of the 3-D reconstruction depends on the accuracy of the correspondences [2]. From the correspondences, we find the projection matrices as well as the 3-D points [3].

Figure 1 shows the relation between a 3-D point and its projected 2-D points such that

$$x = PX \text{ and } x' = P'X. \tag{1}$$

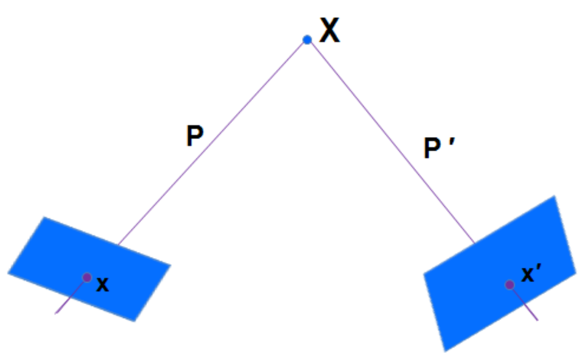


Figure 1. Projection of 3-D point on 2-D images.

where X is a 3-D point, x and x′ are the points in the un-calibrated images and P and P′ are projection matrices.

For the calibrated images, we know a 3-D point of the center of each camera and relation between a 3-D point and its projected point in 2-D image [4]. The reconstruction method using silhouette images is widely used. This method is based on the silhouette constraint that a 3-D object is enclosed in the 3-D cone produced by back-projecting a 2-D object silhouette on an image plane. An approximation of the 3D object shape can be obtained by intersecting such cones using multiple silhouette images. This approximation is called as visual hull. The silhouette volume intersection such as a visual hull model is the most popular method to reconstruct 3-D object shape from multiple 2-D images [5].

The proposed algorithm is designed based on the silhouette images. We reconstruct a more realistic 3-D object by using texture information of the 2-D input images as well as geometry information. Thus, we calculate the texture values of the 3-D points on the object surface from its corresponding points in 2-D images.

In this paper, we propose the 3-D reconstruction method using surface detection. We use the voxel structure based on silhouette images [6]. The voxel is a volumetric element to represent a value of a regular grid in the 3-D space. To reconstruct only surface of the 3-D object, we use the fact that all points on the surface are directly connected to at least one camera.

2. SURFACE RECONSTRUCTION ALGORITHM

Figure 2 illustrates a flow chart of the proposed algorithm. At first, we calibrate the cameras to find the relation between a 2-D point in the image of each camera and its 3-D point in the 3-D space. After that, preprocessing consisting of dilation and erosion more accurately separates the object from the image. The 3-D space is partitioned into small voxels. To detect the 3-D object, each voxel is projected on image plane of each camera. If the voxel is projected on the object region of the 2-D images, it is regarded as a voxel belonging to 3-D object.

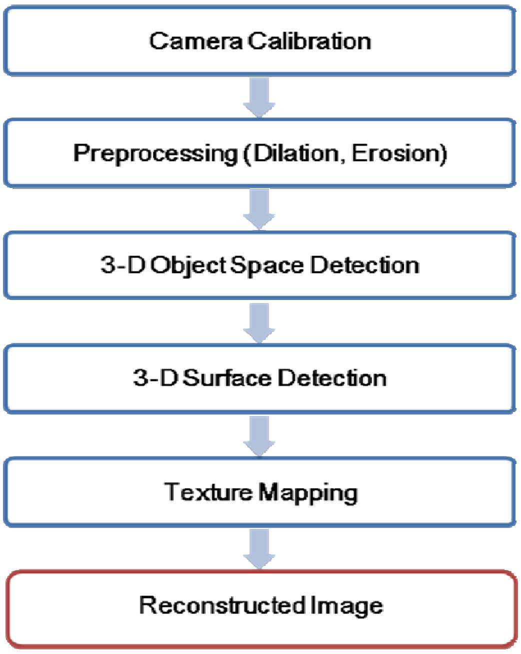


Figure 2. Flowchart of whole process

Figure 3 shows the voxel representation and the projection process from a voxel in 3-D space to the 2-D image planes. Up to this process, the 3-D space of the target object is obtained. And then, we detect the surface of the object using lines between points in 3-D object space and the center of each camera. Finally, we determine the texture value of the point on the detected surface as a median value of the corresponding 2-D points.

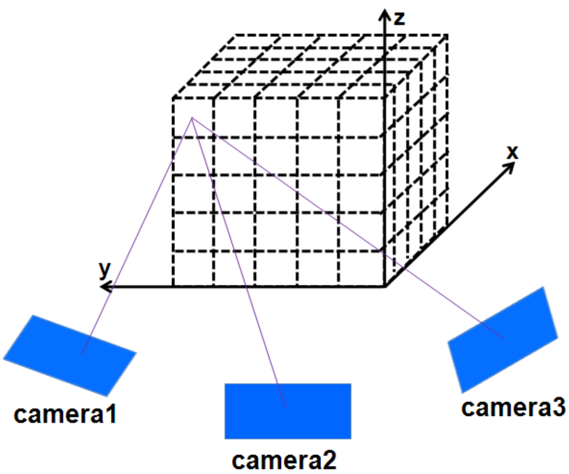
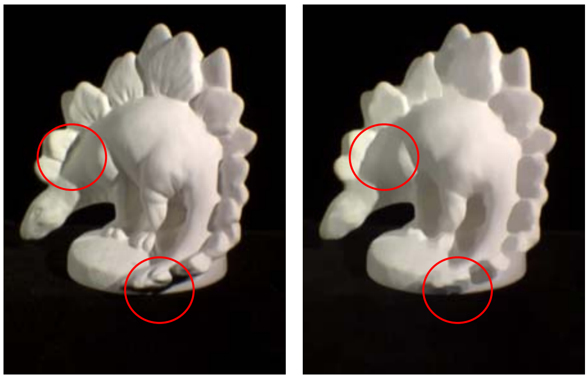


Figure 3. 3-D voxel representation.

2.1 Preprocessing

The test data sets used in this paper can easily separate the object from the image since the background is represented with black-tone colors. However, since some shaded regions of the object also have black-tone colors, they can be regarded belonging to background. To solve above problem, we compensate the shaded regions of the object using image dilation and erosion. Figure 4 shows an example of preprocessing. The circles represent shaded regions of the object in the original image. As you can see in figure 4, the shaded regions are changed with the bright colors in the preprocessed image and can be distinguishable with background. This is helpful to reconstruct the more accurate 3-D object.



(a) (b)

Figure 4. Examples of preprocessing: (a) original image, (b) preprocessed image

2.2 3-D SPACE DETECTION OF OBJECT

The limitation of the 3-D space is helpful to speed up the 3-D reconstruction. Usually the information for the limited 3-D space is available but we can roughly decide the limited 3-D space using camera centers when it is not available. Camera center for each camera is calculated by

$$C_i = -(K_i * R_i)^{-1} * P_i(:,4) \quad \text{for all } i \qquad (2)$$

where the vector, $C_i=[C_x\ C_y\ C_z]'$, represents the center position of i^{th} camera, K_i is the 3x3 matrix representing an internal parameter of the camera, R_i is the 3x3 matrix representing a rotation parameter of the camera, $P_i(:,4)$ represents the last column of the camera matrix P_i.

To detect the 3-D space of an object, points in 3-D space are projected on the image plane of each camera. For that, we use the projection matrix for the calibrated images [2]. In this process, we use the preprocessed images in figure 4 (b) instead of the original images. If a certain point in the 3-D space is projected on the object region of all 2-D images, it is regarded as a real 3-D point of the object. If not, that point is regarded as an outer point of the object. As the number of cameras increases, we can get the more accurate 3-D space of the object.

2.3 3-D SURFACE DETECTION

Up to this process, we obtained the 3-D space of an object. But this 3-D space of the object includes not only the surface points but also the internal points of the object. To get the surface points only, we make a line between a point in the 3-D space of the object and a camera center. This line equation is defined by

$$X = [X-C] * t + C. \quad (3)$$

Where the vector, X=[x y z]′, represents the point in the 3-D space of the object. C means the position of the camera center and t represents an arbitrary scalar value.

Figure 5 shows direct lines between the camera centers and one point in the detected 3-D space and they are used for a surface detection. In the same manner, we have to define line equations for all other points.

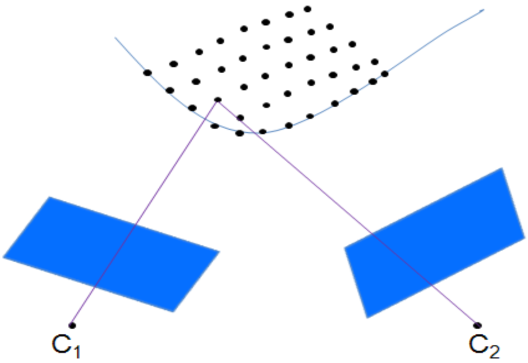


Figure 5. Line equation for accurate range of object

If the connected lines between one point in the object space and all camera centers include certain other point in the object space, we can know that this point is not a surface point. Thus, we exclude this point from the candidates of surface points. If we apply this process to all views, we get accurate surface. Figure 6 shows the process of surface detection. In Figure 5, A, B, C, and D are points in the space of object. The line $\overline{AC_1}$ which connects point A and camera center C_1 does not meet any points in space of object. We can regard point A as the surface point of the object. Point B and C also have more than one direct line. So we can point B and C as the surface points of the object. However, line $\overline{DC_1}$, meets points in a space of object. Line $\overline{DC_2}$ and line $\overline{DC_3}$ also meet points in space of object. Therefore we can know that point D is not a surface point.

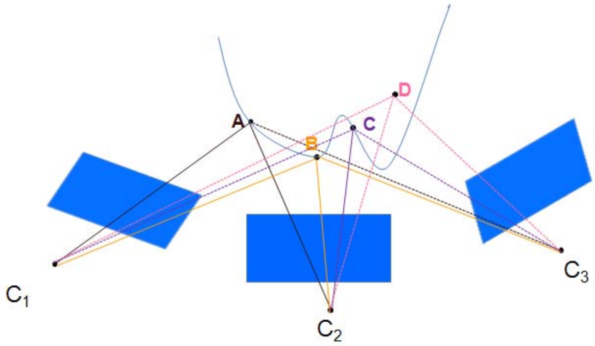


Figure 6. Surface detection. C_1, C_2, and C_3 are camera centers. For visibility checking, we find points which are directly connected to at least one camera center. A: two direct line, one indirect line → surface, B: three direct lines → surface, C: one direct line, two indirect lines → surface, D: three indirect lines → not surface

2.4 TEXTURE MAPPING

After getting surface, we should map the proper values to surface points. In this process, we use the original images and line between obtained surface point and camera center. After we connect the surface point to all camera centers, we select the cameras that are connected to the surface point directly. Surface point is projected on the original image of selected camera to get the texture values. Then we sort these texture values. We map the median value of these sorted texture values to 3-D surface point. After we apply this process to all surface points, we can get the reconstructed image.

3. EXPERIMENTAL RESULT

We tested our algorithm using “DinoSparseRing” data set and “TempleSparseRing” data set which include 16 views sampled on a ring around the object [7]. The size of each image is 640 $\times$ 480. To render 3-D image, we used voxel representation. When we experiment our proposed algorithm, we used 3-D range given from data set. Figure 7 shows the experimental results for “DinoSparseRing”.

(a) (b)

(c) (d)

Figure 7. Results for “DinoSparseRing” data set: (a) 1st view, (b) reconstructed image for (a), (c) 9th view, (d) reconstructed image for (c)

Figure 8 illustrates the original and reconstructed images for “TempleSparseRing” data set.

(a) (b)

(c) (d)

Figure 8. Results for “TempleSparseRing” data set: (a) 1st view, (b) reconstructed image for (a), (c) 7th view, (d) reconstructed image for (c)

From the experimental results, we confirmed that the shaded parts of the object are compensated through a preprocessing and the accurate 3-D shapes were obtained by a surface detection. In addition, we map the textures for the reconstructed from the original images. The reconstructed 3-D object looks similar to original 2-D images.

4. CONCLUSION

In this paper, we proposed a 3-D reconstruction using surface detection. We use a preprocessing consisting of the image dilation and erosion process to compensate the shaded region in the object. We get the 3-D space of the object by checking the occupancy of each voxel. We use the line equation from each camera center to the points of detected object space to find the surface points of the object. Finally, we fill the surface points with the median value of their projected 2-D points on the original images. The experimental results demonstrate that the reconstructed images using the proposed 3-D reconstruction method are subjectively natural as original images.

5. ACKNOWLEDGMENT

"This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute for Information Technology Advancement)" (IITA-2009-C1090-0902-0017).

REFERENCES

1. M. Grum and A. G. Bor, “Multiple Image Disparity Correction for Scene Representation,” Proc. of IEEE International Conference on Image Processing, p.209-212 (2008).
2. R. Hartley and A. Zisserman, “Multiple view Geometry in Computer Vision,” Cambridge University Press 2000.
3. Y. Lu, J. Z. Zhang, Q. M. Jonathan Wu, L. Ze-Nian, “A Survey of Motion-Parallax-Based 3-D Reconstruction Algorithm,” Proc. of IEEE Transactions on System, Man, and Cybernetics-Part C: Applications and Reviews, Vol. 34, No 4, (2004)
4. D. A. Forsyth, J. Ponce, “Computer Vision a Modern Approach,” Pearson Education International 2003
5. T. Matsuyama, X. Wu, T.Rakai, S. Nobuhara, “Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualization for 3D video,” Proc. of Computer Vision and Image Understanding , Vol 96, Issue 3, pp. 393-434(2004)
6. K.M. Cheung, T. Kanade, J. Bouguet, and M. Holler, “A real time system for robust 3D voxel reconstruction of human motions,” Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 714-720 (2000).
7. Middlebury “DinoSparseRing” data set and “TempleSparseRing “ data set available at: http://vision.middlebury.edu/mview/data/