

Panoramic Mesh Model Generation for Indoor Environment Modeling*

Wonwoo Lee and Woontack Woo

GIST U-VR Lab., Gwangju 500-712, S.Korea

{wlee, wwoo}@gist.ac.kr

Abstract

In this paper, we propose a panoramic mesh model generation method. There are complex problems in modeling multi-view range data, such as registration and triangulation of 3D point cloud. The proposed method generates a mesh model of indoor environment from an unorganized 3D point cloud. First, we divide input point cloud into several non-overlapping areas. Then, we project each area to a virtual camera and triangulate it on virtual 2D image plane. We back-project each 2D mesh to 3D space. Finally, we generate panoramic mesh model of the environment by merging adjacent mesh models. Since 3D point cloud is triangulated on 2D plane, we avoid complex 3D triangulation. The proposed method is useful in modeling large area such as indoor environment where range sensors are not suitable for data acquisition.

Key words: Virtual Environment, Mesh modeling, Indoor environment.

1. Introduction

Modeling virtual environment from the real world is one of virtual reality applications and there is increasing interest in this field. It provides immersive feeling to users and is used in education, industry and entertainment, etc. It is complex and time consuming work to create 3D model of large environment with modeling tools, such as 3DS-Max and Maya. Modeling virtual environment from the real world has several advantages. It generates realistic models of the world since it uses images of the real world as textures. It exploits existing 3D information of the environment so that modeling process can be automatic without manual input.

There have been researches on modeling virtual environment from the real environment. One approach is using range sensors to obtain range data from the environment [1][2]. The environment is scanned and 3D information of the environment is obtained as point clouds. Textures obtained from the photos of the environment are mapped to the generated model for realism. Another approach is reconstructing environment

models by extracting 3D information from multi-view images [3]. 3D structure of the environment is generated from relationship among the images. Panoramic images of the environment are also used in modeling [4][5]. Range sensors provide accurate data, however they are designed for scanning small objects in near distance. Thus, it is not convenient to model large area.

In this paper, we propose a panoramic mesh model generation method from unorganized 3D point cloud. First, we obtain range data of indoor environment from multiple views using multi-view camera. By integrating multi-view range data, a registered 3D point cloud is generated as input to the algorithm. Next, we divide the point cloud into several non-overlapping areas. Then, we project each area to each virtual camera. We generate mesh model of projected area through sampling and triangulation on 2D image plane. Finally, we back-project each 2D mesh to 3D space and generate panoramic mesh model of the indoor environment by merging adjacent mesh models. The proposed algorithm provides a simple way to generate panoramic model of indoor environment. It simplifies complex registration problem through the proposed sampling process. There is no need to remove duplicated points in registration step. It can also generate mesh models in 3D by triangulating points on 2D plane. The proposed method is useful in modeling large environment, where range finders are not suitable for data acquisition.

The rest of this paper is organized as follows. We explain modeling process and show experimental results in section 2 and 3, respectively. Conclusions and future works are given in section 4.

2. Panoramic Mesh Model Generation

2.1. Data Acquisition

In data acquisition step, we obtain the data of indoor environment as point clouds using multi-view cameras. To obtain accurate data, we calibrate the multi-view camera and we calculate the intrinsic parameters. From the images captured by multi-view camera, we calculate disparity map. The data obtained from a viewpoint has its own reference coordinate system. We register all data

* This research was supported in part by University IT Research Center Project, and in part by ICU Digital Media Lab.

to locate them in a common reference coordinate system [6]. Since there are overlapped areas among images, there are several points that correspond to one point in real environment. Their coordinates should be identical ideally. However, they are not identical due to calibration errors and noises in resulting range data. Consequently, the registration step results in errors. These duplicated points are removed in sampling process to simplify registration.

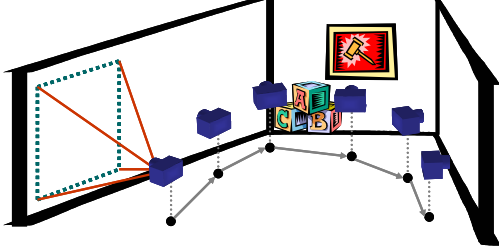


Figure 1. Data acquisition from multiple viewpoints

2.2. Sampling

In sampling step, we divide input point cloud to several sub-point clouds and create new data set from each sub-point cloud through interpolation. Then we approximate the coordinates of the point clouds to reduce errors.

First, we divide input point cloud to N sub-point clouds based on camera's field of view. For k th sub-point cloud S_k , we apply rotation and translation to transform S_k from absolute coordinates to virtual camera coordinates. Then, we project a point (x_i, y_i, z_i) in S_k onto the image plane of the virtual camera as shown in Figure 2.

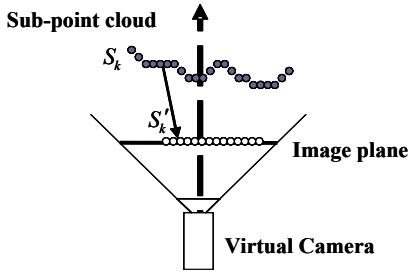


Figure 2. Projection of a sub-point cloud

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = A \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{camera} \quad \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{image} = \begin{bmatrix} x'/z' \\ y'/z' \\ 1 \end{bmatrix} \quad A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Equation (1) describes the projection. From the point (x, y, z) in camera coordinates, we get a point (u, v) on image plane of the camera. As the result of projection, we obtain a point cloud S'_k on 2D image plane. A is the matrix consisting of the internal parameters. f_x and f_y are focal lengths of the camera in horizontal and vertical direction, respectively. c_x and c_y are the coordinates of the principal point of the camera.

After we obtain a point cloud S'_k on the image plane, we create a grid on the virtual camera's image plane. The resolution of the grid is identical to that of images used in data acquisition step. If a point on the grid is inside the 2D point cloud projected from 3D space, the point is valid. Let the set of valid points on the grid be G'_k .

When the set of valid points on the grid is determined, we form a new point cloud in 3D space. We consider a point on the grid as an analogue of a pixel in the image. For a valid grid point P' , there is a point P_{3D} in 3D space which corresponds to P' . The coordinates and color values of P_{3D} can be calculated from the four nearest points, P'_1, P'_2, P'_3, P'_4 . This idea is shown in Figure 3. The coordinates of $P_{3D}(x, y, z)$ are interpolated from $P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3)$, and $P_4(x_4, y_4, z_4)$ which are corresponding points to P'_1, P'_2, P'_3, P'_4 . The color value $P_{3D}(r, g, b)$ is interpolated from $P_1(r_1, g_1, b_1), P_2(r_2, g_2, b_2), P_3(r_3, g_3, b_3), P_4(r_4, g_4, b_4)$ through equation (2). l_i is the distance from P to the i th nearest point. The nearer point has the larger weight factor. As a result of interpolation, we get a new point cloud G_k in 3D space

$$\begin{aligned} x &= \frac{1}{3} \sum_{i=1}^4 \left(1 - \frac{l_i}{l_{total}}\right) x_i & r &= \frac{1}{3} \sum_{i=1}^4 \left(1 - \frac{l_i}{l_{total}}\right) r_i \\ y &= \frac{1}{3} \sum_{i=1}^4 \left(1 - \frac{l_i}{l_{total}}\right) y_i & g &= \frac{1}{3} \sum_{i=1}^4 \left(1 - \frac{l_i}{l_{total}}\right) g_i \\ z &= \frac{1}{3} \sum_{i=1}^4 \left(1 - \frac{l_i}{l_{total}}\right) z_i & b &= \frac{1}{3} \sum_{i=1}^4 \left(1 - \frac{l_i}{l_{total}}\right) b_i \end{aligned} \quad (2)$$

Where, $l_{total} = \sum_{i=1}^4 l_i$

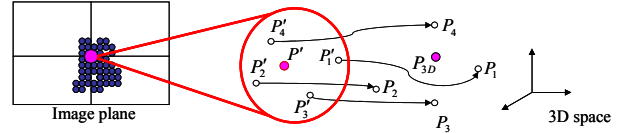


Figure 3. Interpolation

There exist errors in the coordinates of points in the point cloud obtained from a camera. Even though a set of points is on a plane in real 3D space, it has fluctuation in the acquired data. These errors make the appearance of the model worse. To reduce the errors, we calculate the least mean-square plane of a set of points in G_k and project the points to the plane to recalculate their 3D coordinates.

We compare the z coordinates of two points $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ which correspond to the adjacent points on the grid in G'_k . If the absolute value of the difference, $|z_1 - z_2|$, is smaller than a threshold value τ , we assume that they are on the same plane. After a set of the points on the same plane is found, we calculate a plane Ω which minimizes the least mean-square error. Then the coordinates of the points are recalculated by projecting them onto Ω . If the cardinality of the set of points assumed to be on the same plane is less than 3, we don

not approximate the coordinates of the points in the set. The pseudo code of this process is shown in Table 1.

Table 1. Pseudo code of approximation

```

While (There remains any point in  $G_k'$ )
  Choose a point  $p$  in  $G_k'$ 
  Create new set  $V$ 
   $V = V \cup \{p\}$ 
  For all points in  $G_k'$  except  $p$ 
    Select a point  $p'$  in  $G_k'$ 
    If  $(|z_1 - z_2| < \tau)$ 
      THEN  $V = V \cup \{p'\}$ ,  $G_k' = G_k' - \{p'\}$ 
  End While

For all partitions
  If cardinality is less than 3
    Continue
  Calculate a plane  $\Omega$ 
  For all points in the partition
    Project a point onto  $\Omega$ 

```

2.3. Triangulation

To generate mesh model from the sampled point cloud, we need to create connectivity information among the points. There is one to one mapping between G_k and G_k' and we utilize this mapping. By triangulating the set of valid grid points G_k' , we can generate 2D mesh on image plane.

We apply the connectivity information of G_k' to the points of G_k . Thus, it is possible to triangulate the sampled point cloud in 3D space. Through this, we can bypass complex 3D triangulation. We assume that three points which forms a triangle on the grid of image plane have high possibility that they form a triangle in 3D space too. The grid points are triangulated in consistent way based on this assumption. Figure 4 depicts the triangulation method. By triangulating all the sampled point cloud, we obtain a mesh model from a sub-point cloud.

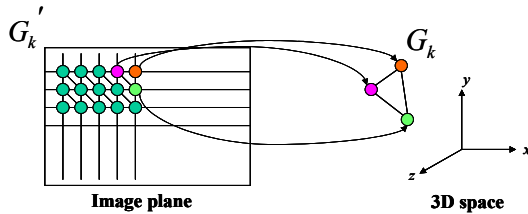


Figure 4. Triangulation in 2D and 3D

Since we divide the input point cloud and generate mesh models of sub-point clouds, there exists a gap between two adjacent mesh models. To obtain one panoramic model of the indoor environment, we merge all sub-models to one mesh model. We triangulate the points on boundaries of models to merge two adjacent mesh models. The points on the boundary of the model are the points corresponding to the leftmost and rightmost grid points. Since the number of points on boundary is not identical in both models, simple triangulation results in

distortion at the boundary. To avoid distortion after triangulation, we connect the points which have similar height in 3D space. Then, we triangulate the points the points which have no connection. This merging idea is shown in Figure 5.

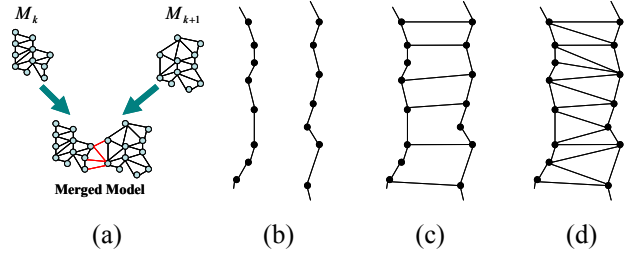


Figure 5. Mesh merging step (a) The idea of mesh merging (b) The points on boundary (c) Connecting the points in the same height (d) Triangulated points on boundary

3. Experimental Results

We applied the proposed method to synthetic data which is ideal data set and real data which contains noise to verify our modeling method.

Figure 6 shows partial modeling results of synthetic data. Figure 6(a) shows input point cloud of synthetic indoor environment. Figure 6 (b) and Figure 6(c) describe valid grid points and triangulation result of the valid grid points on image plane, respectively. The points in the dotted rectangle in Figure 6(a) are partitioned to two sub-point clouds. Then, they are sampled and triangulated. Partial model is rendered with point set in Figure 6(d). There is visible difference between the modeled area and the area consisting of only points. There still exists a gap in Figure 6 (d) because they are not merged yet.

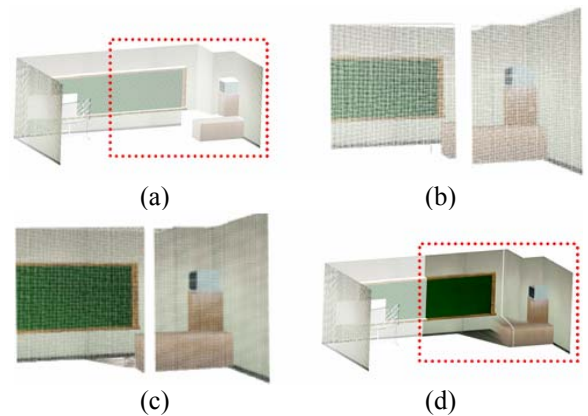


Figure 6. Partial modeling results of synthetic data (a) Input point cloud (b) Sampled points in 2D image plane(c) Triangulation result of sampled points (d) Generated mesh model in 3D space

To obtain 3D input point cloud, we used Digiclops which is a multi-view camera manufactured by PointGrey [8]. The size of image is 640x480. Figure 7

shows the modeling result for one scene. Two point clouds are registered from two point clouds as shown in Figure 7(a). Figure 7(b) depicts the sampled point cloud. The original point cloud has vacancy where the 3D information is not obtained because of the limitation of disparity estimation. Disparity estimation has high possibility to fail in homogeneous area. The sampled point cloud has similar vacancy since the coordinates of the points are interpolated. Figure 7(c) shows the mesh model generated from the sampled point cloud. There are a lot of points so the mesh is dense. The holes in Figure 7(b) are filled with triangles. Figure 7(d) shows textured model. Each point has its own color and the colors of three points of a triangle are interpolated by Gouraud shading. Since homogeneous areas usually result in holes, the filled holes do not look weird even though the color is interpolated.

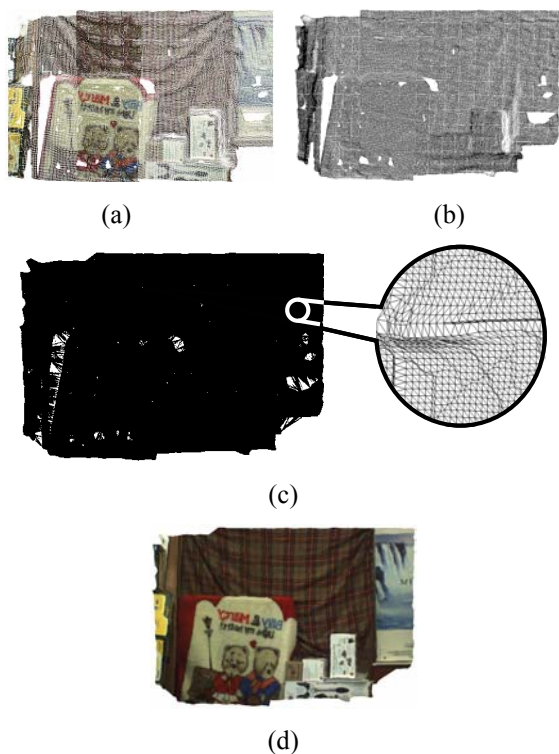


Figure 7. Modeling result for one scene (a) Raw point cloud (b) Sampled point cloud (c) Mesh model (d) Textured model

Figure 8 shows the result of the least mean-square approximation. The models shown in Figure 8(a) and Figure 8(c) have fluctuation. This distortion affects user's visual feeling significantly. The approximation results are shown in Figure 8(b) and (d). After approximation, the surface of the model, such as the area inside the circle drawn in line, becomes flat. However, this approximation can create defects. Two sets of points are assumed to be on different plane even though they are on the same plane in real environment, if they have errors larger than the specified threshold value. This defect is shown in the areas inside dotted circle.

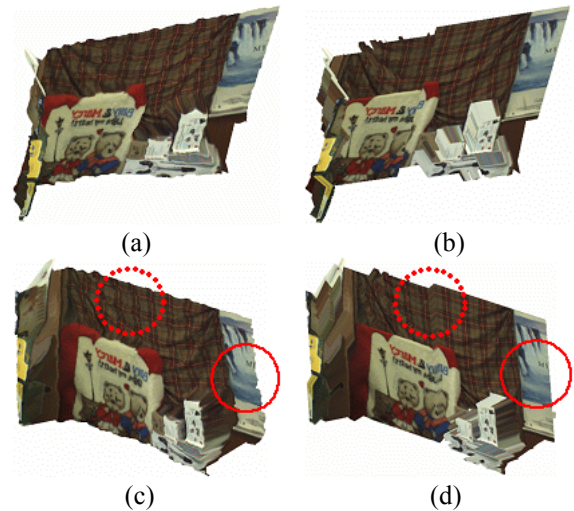


Figure 8. The least mean-square approximation results (a) Before (Viewpoint 1) (b) After (Viewpoint 1) (c) Before (Viewpoint 2) (d) After (Viewpoint 2)

Figure 9 shows the mesh merging process. Two adjacent mesh models are shown in Figure 9(a) and (b). Figure 9(c) and (d) show a magnified view of the gap between two models. As shown in Figure 9(c), we find a set of points on boundary in each model. They are triangulated as proposed and a part of the result is shown in Figure 9(d). Figure 9(e) and (f) shows magnified view of the scene. The gap between two models, shown in Figure 9(e), are removed after mesh merging, as shown in Figure 9(f).

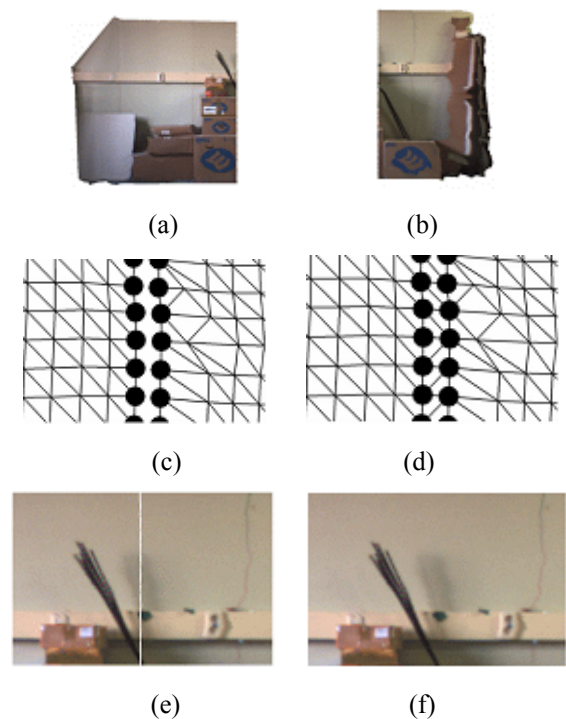


Figure 9. Mesh merging (a) Model 1 (b) Model 2 (c) Points on boundary (d) Triangulated points on boundary (e) Before mesh merging (f) After mesh merging

We chose a room as an indoor environment and applied our method to the room. In data acquisition step, we obtained a point cloud for a wall of the room. The room has four walls and four point clouds are generated.

Figure 10 shows a modeling result of a wall. As shown in Figure 10(b), input point cloud is partitioned to three sub-point clouds according to the virtual camera's field of view. In experiment, we used 45 degree. Partial models are generated from each sub-point cloud. Three partial models were merged to one model. Figure 10(c) and Figure 10(d) depict the partial models and the merged model. In Figure 10(a), there exist gaps in the point cloud and color variance. Since lighting condition varies according to the position of the camera in the room, the same point can have different color in overlapped area of images. After modeling, the color difference is reduced but still exists.

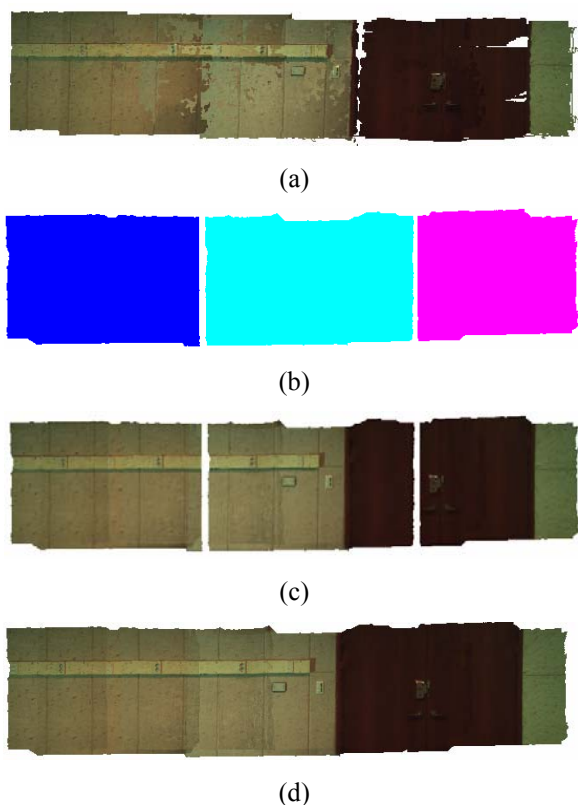


Figure 10. Modeling result of a wall (a) Input point cloud (b) Partitioned point cloud (c) Mesh model of each sub-point cloud (d) Integrated mesh model after mesh merging

Figure 11 shows the modeling result of a room. Whole environment is shown in Figure 11(a). Figure 11(b), (c) and (d) are the rendered scenes of the generated model from different viewpoints.

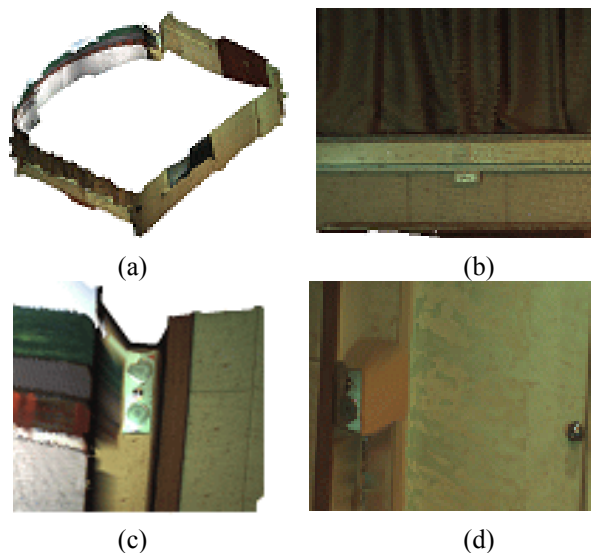


Figure 11. Modeling result of the room (a) Generated model of the room (b) Viewpoint 1 (c) Viewpoint 2 (d) Viewpoint 3

Table 2 shows the number of points and faces of the reconstructed model. The number of points is reduced by sampling. However, there are a lot of points and faces in reconstructed model.

Table 2. The number of points and faces of the reconstructed model

	Input points	Reconstructed model	
		Vertices	Face
Wall 1	932801	537956	1071305
Wall 2	1144816	640868	1277335
Wall 3	1354335	705602	1407042
Wall 4	1194347	811558	1618636

4. Conclusion & Future Works

In this paper, we propose a mesh model generation method from an unorganized point cloud for indoor environment modeling. The proposed algorithm provides a simple way to generate panoramic model of indoor environment. It simplifies the complex registration problem through the sampling process. It can also generate a mesh model in 3D by triangulating points on 2D plane. The proposed method is useful in modeling large environment, where range finders are not suitable for data acquisition. There still remains several works to improve our method. The generated mesh model consists of too many triangles. Since fewer triangles are preferred in practical usage, mesh simplification is inevitable. Current data structure of mesh model uses color per vertex in which every vertex has its color value. To maintain realism after simplification, we need to generate texture map from color information of each vertex. In addition, color smoothing in overlapped area is necessary to generate more realistic model.

References

1. Vitor Sequeira, João Goncalves, M. Isabel Ribeiro, "3D Reconstruction of Indoor Environments", *ICIP96*, pp.405-408, Lausanne, Switzerland, 1996
2. Y. Sun, J.K. Paik, A. Koschan, and M.A. Abidi, "3D reconstruction of indoor and outdoor scenes using a mobile range scanner", *Pattern Recognition*, vol.3, pp 653-656, 2002
3. Johnson, S. Kang, "Registration and Integration of Textured 3-D Data" Tech. report CRL96/4, Digital Equipment Corporation, Cambridge Research Lab, 1996.
4. McMillan, L., G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", *Proceedings of SIGGRAPH 95*, pp. 39-46, 1995
5. H.Y. Shum, M. Han, and R. Szeliski, "Interactive construction of 3d models from panoramic mosaics", In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp 427-433, 1998.
6. S. Kim, K. Kim, and W. Woo, "Depth based Registration for Image based Virtual Environment Generation", *Workshop on Image Processing and Image Understanding on Korea*, vol. 1., pp. 154-159, 2004
7. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points", *Computer Graphics, Proceedings of SIGGRAPH 92*, 26(2):71-78, 1992.
8. PointGrey, <http://ptgrey.com>