Free-Viewpoint Rendering from Omnidirectional Video Using a Deformable 3-D Mesh Model

Hiroyuki Koshizawa[1], Tomokazu Sato[1], and Naokazu Yokoya[1]

Graduate School of Information Science,
Nara Institute of Science and Technology (NAIST),
8916-5 Takayama, Ikoma, Nara, Japan
{tomoka-s, yokoya}@is.naist.jp

Abstract. This paper proposes a method to render free viewpoint images using a deformable 3-D mesh model and omni-directional video. In the proposed method, the 3-D mesh is placed in front of the virtual viewpoint and deformed by using the pre-estimated omni-directional depth maps that are selected based on the position and posture of the virtual viewpoint. Although our approach is basically based on the MBR approach that can render the geometrically correct virtualized world, in order to avoid the hole problem, we newly employ a viewpoint-dependent deformable 3-D model instead of the use of the unified 3-D model that is generally used in MBR approach. In the experiment, free viewpoint images are generated from omni-directional video captured by an omni-directional multi-camera system to show the feasibility of the proposed method for walk-through applications in the virtualized environment.

1 Introduction

One trivial goal of representation and modeling of large-scale 3-D environment is development of a high quality virtualized world that is based on the real environment. In recent years, virtualized worlds based on the real world have been released from Microsoft (Virtual Earth) and Google (Google Earth). These virtualized worlds now realizes virtual sightseeing, navigation, and will be used in wider range applications such as entertainment, digital archiving and education. However, in the current version of the virtualized worlds, details of the real world are omitted due to the cost of the 3-D modeling. Thus, the reality of the virtualized world has still not reached to the sufficient level for some applications.

There are many researches that try to automatically construct a virtualized real world in order to reduce human cost for modeling. Most of these researches can be categorized based on the rendering policy of the virtualized world. One is model based rendering (MBR) and the other is image based rendering (IBR). MBR methods render the virtualized world based on explicit 3-D models. The key problem in MBR is how to automatically generate explicit 3-D models of the real world. There are several ways: shape from shading, silhouette, focus and

defocus, motion, stereo. In these categories, combination of motion and stereo are comparably familiar with outdoor environments. M. Goesele et. al [1] have developed a method that estimates a geometry of an outdoor environment from community photographs. P. Merrell et. al [2] proposes the fast algorithm to estimate geometry of outdoor environment using captured images taken by the car-mounted camera system. These methods generate a 3-D model of a target environment using video images. Car mounted laser range finder is also other option for scanning outdoor environments [3, 4]. Although both vision-based and laser-based methods realize (semi-)automatic 3-D modeling of large outdoor environments, estimated 3-D models cannot be directly employed for the virtual environment because these models have many holes caused by occlusions and estimation errors. Even by the state-of-the-art methods, complete 3-D modeling for a large outdoor environment is very difficult due to un-visible and un-measurable parts from the observation points.

IBR method renders the images of the virtual viewpoint without explicit 3-D geometry [5]. Generally, IBR method can render realistic images only when the virtual viewpoint is set at near from the original viewpoint. However, rendered image is easily distorted when the virtual viewpoint is set at far from the original viewpoint. In order to avoid distortion problem, some methods employ view-dependent implicit geometry [6, 7]. These methods determine the pixel value of generating image by estimating depth of each pixel from the virtual viewpoint using the photo-consistency. However, on-demand depth estimation is not familiar with walk-through applications because large computational cost is needed for each loop of viewpoint rendering.

In this paper, we propose a method that renders free viewpoint images using 3-D mesh model that is deformed by using pre-estimated depth maps for original viewpoints. In our approach, by selecting and merging appropriate depths and textures from several viewpoints, realistic images without holes and distortions can be generated even when the virtual viewpoint is set at the distancing place from the original viewpoint. Contributions of this paper are that (1) geometry of the scene for the virtual view point can immediately be recovered by fitting a deformable mesh model to pre-estimated omni-directional depth maps for original viewpoints, (2) any holes appear in generated images by using the 3-D mesh model that is deformed as optimal shape for the scene structure.

The rest of this paper is organized as follows. In Section 2, the method for free-view point rendering using deformable 3-D mesh model is described. Section 3 shows the experimental results for walk through applications in the virtualized world, and Section 4 summarizes this paper.

2 Free-viewpoint rendering using view-dependent 3-D mesh model

Fig. 1 shows a flow diagram of the proposed method for free-viewpoint rendering. In this research, in order to render the images for arbitrary directions in the virtualized world, input images are captured by an omni-directional multi-camera

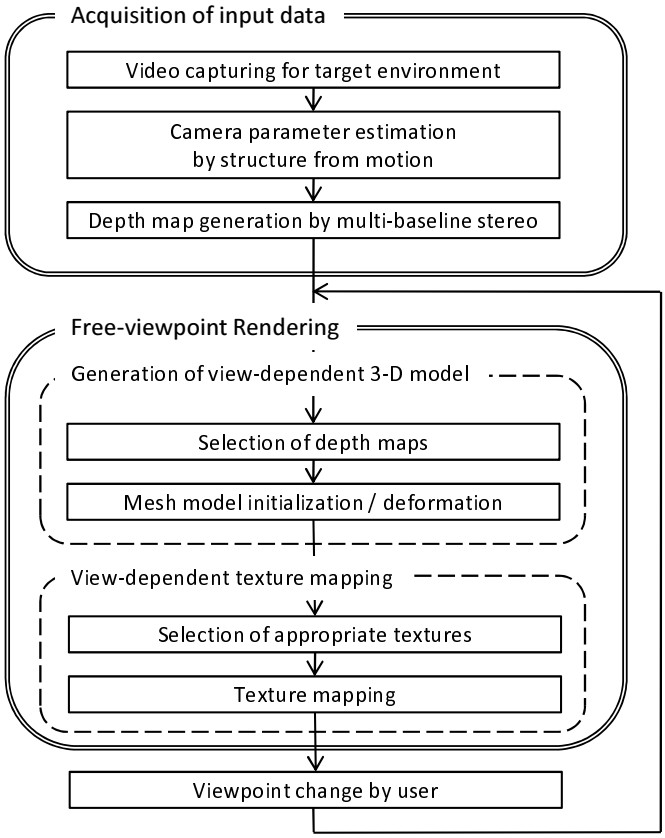


Fig. 1. Flow diagram of proposed method.

system (OMS). By tracking feature points on the captured images, camera parameters of the OMS are estimated using structure-from-motion algorithm designed for OMS [8]. After that, depth map for each input image is also estimated using multi-baseline stereo [9].

In the rendering stage, 3-D mesh model is placed in front of the virtual viewpoint and is deformed so as to minimize an energy function that expresses consistency of the estimating view-dependent depth map and pre-estimated depth maps for original viewpoints. After deformation of the mesh, appropriate texture for each polygon is mapped onto the deformed mesh from the original images.

2.1 Generation of view-dependent 3-D model

As shown in Fig. 2, before starting the rendering stage, 3-D mesh model is initially placed in front of the virtual viewpoint as plane model whose distance from the virtual view-point is F and is parallel to the image plane of the virtual

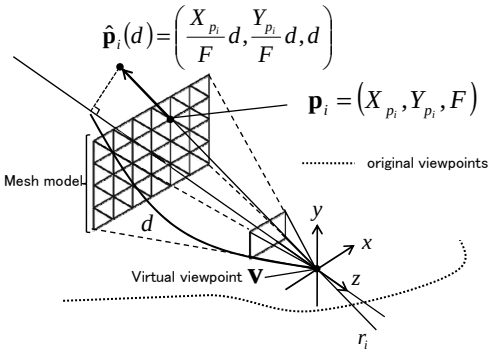


Fig. 2. 3-D coordinates of vertexes on deformable mesh model.

camera. After mesh initialization, 3-D mesh model is deformed using energy function that is depending on the position and posture of the virtual viewpoint.

Definition of energy function Each vertex on the mesh model is moved so as to minimize the energy function that expresses consistency of the depth data from original viewpoints. As shown in Fig. 2, destination position $\hat{\mathbf{p}}_i$ of the i-th vertex $\mathbf{p}_i$ in mesh deformation is constrained on the straight line connecting the virtual viewpoint $\mathbf{v}$ and the initial position $\mathbf{p}_i$. The energy $E_i(d)$ for the i-th vertex is defined as the function of the depth d in the virtual viewpoint $\mathbf{v}$:

$$E_i(d) = \frac{\sum_{j\in\mathbf{f}_i} w_{ij}(\hat{\mathbf{p}}_i(d))D_{ij}(d)^2}{\sum_{j\in_{\mathbf{f}_i}} w_{ij}(\hat{\mathbf{p}}_i(d))}, \tag{1}$$

$$D_{ij}(d) = \begin{cases} e(\mathbf{c}_{ij},\hat{\mathbf{p}}_i(d)) - |\hat{\mathbf{p}}_i(d)-\mathbf{c}_{ij}| \\ \qquad ; e(\mathbf{c}_{ij},\hat{\mathbf{p}}_i(d)) > |\hat{\mathbf{p}}_i(d)-\mathbf{c}_{ij}| \\ 0 \qquad ; otherwise \end{cases}, \tag{2}$$

$$w_{ij}(\hat{\mathbf{p}}_i(d))) = \begin{cases} \theta_{ij}(d)^{-1} = \arccos\left(\frac{(\hat{\mathbf{p}}_i(d)-\mathbf{c}_{ij})\cdot(\hat{\mathbf{p}}_i(d)-\mathbf{v})}{|\hat{\mathbf{p}}_i(d)-\mathbf{c}_{ij}||\hat{\mathbf{p}}_i(d)-\mathbf{v}|}\right)^{-1} \\ \qquad ; e(\mathbf{c}_{ij},\hat{\mathbf{p}}_i(d)) > |\hat{\mathbf{p}}_i(d)-\mathbf{c}_{ij}| \\ 0 \qquad ; otherwise \end{cases}, \tag{3}$$

where $\mathbf{f}_i = (f_{i1}, f_{i2}, \cdots, f_{iN})$ is the frame indexes of the original viewpoints $(\mathbf{c}_{i1}, \mathbf{c}_{i2}, \cdots, \mathbf{c}_{iN})$ that are used to compute energy function E_i. The function $e(\mathbf{c}_{ij}, \hat{\mathbf{p}}_i(d))$ returns the depth value for direction $\hat{\mathbf{p}}_i(d)$ on the original viewpoint $\mathbf{c}_{ij}$ as shown in Fig. 3. $w(\hat{\mathbf{p}}_i(d))$ is weighting function for the original viewpoint $\mathbf{c}_{ij}$ that is selected for the vertex $\mathbf{p}_i$ and this function is defined as inverse of the angle $\theta_{ij}(d)$ [degree] between the lines: the line connecting from $\mathbf{c}_{ij}$ to $\hat{\mathbf{p}}_i(d)$ and the line connecting from $\mathbf{v}$ to $\hat{\mathbf{p}}_i(d)$.

As defined in Eqs. (2) and (3), element energy D_{ij} and its weight $w(\hat{\mathbf{p}}_i(d))$ for the j-th original viewpoint are set 0 if $e(\mathbf{c}_{ij}, \hat{\boldsymbol{p}}_i(d)) > |\hat{\mathbf{p}}_i(d) - \boldsymbol{c}_{ij}|$. This condition is satisfied when the position $\hat{\mathbf{p}}_i(d)$ is occluded by other objects from

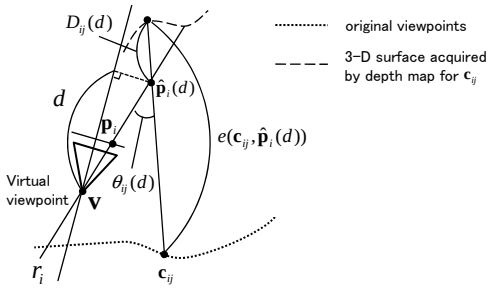


Fig. 3. Parameters used for computing energy.

the j-th original viewpoint $\mathbf{c}_{ij}$. It should be noted that if most of the original viewpoints satisfy the above occluding condition, energy E_i will be unstable because number of original viewpoints for energy determination is too few. Thus, the depth d is skipped in the energy minimization process when the number of original viewpoints that are used for energy determination is M or lower for depth d.

Selection of depth map The depth maps that are used for computing the energy E_i are selected by using the distance between the original viewpoint $\mathbf{c}_{ij}$ and the ray r_i from the virtual viewpoint $\mathbf{v}$ to the position of the vertex $\mathbf{p}_i$. More concretely, first, the Euclidian distance l_{ij} from the ray r_i to the original viewpoint $\mathbf{c}_{ij}$ is computed for each j. The top N nearer original viewpoints $(\mathbf{c}_{i1}, \mathbf{c}_{i2} ..., \mathbf{c}_{iN})$ from the ray r_i and their associated depth maps are then selected using the distance l. These viewpoints are selected based on the idea that there should exist comparably fewer occluders along the ray r_i than the other viewpoints.

Deformation of mesh model At the initial time of the rendering stage, the depth value d that minimizes the energy E is searched for all the given range $[d_{min}, d_{max}]$. Each vertex $\mathbf{p}_i$ is moved to the position $\hat{\mathbf{p}}_i(d_{E_{min}})$ where the energy E_i is minimized. $d_{E_{min}}$ is determined as follows;

$$d_{E_{min}} = \underset{d_{min} \leq d \leq d_{max}}{\operatorname{argmin}} \; E_i(d). \tag{4}$$

Except the initial time of the rendering stage, the 3-D mesh model generated for the previous virtual viewpoint can be used as an initial mesh model, and is used to limit the searching range for depth value. This limitation for the depth search can decrease computational cost of view-dependent model generation for each frame. Fig. 4 illustrates the limited range for depth search when the virtual viewpoint is moved forward. The surface (A) in this figure illustrates the mesh model that is generated for the previous camera position. In the proposed method, searching range of the depth is limited inside the surfaces (B) and (C) that are placed around the surface (A). Concretely, for the i-th vertex $\mathbf{p}_i$, the

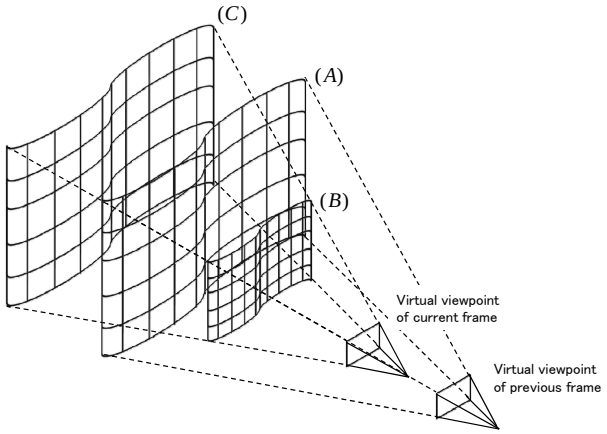


Fig. 4. Limited range for depth search.

intersecting point $\tilde{\mathbf{p}}_i$ of the previous mesh model and the ray r_i is firstly computed. By using the depth $\tilde{d}$ from the virtual viewpoint of the current frame to the intersecting point $\tilde{\mathbf{p}}_i$, depth $d_{E_{min}}$ is determined using the searching range $[(1-R_d)\tilde{d}, (1+R_d)\tilde{d}]$.

$$d_{E_{min}} = \underset{(1-R_d)\bar{d} \leq d \leq (1+R_d)\bar{d}}{\operatorname{argmin}} E_i(d). \tag{5}$$

It should be noted that if occluding edges exist in the scene, this scheme cannot work well because true depth will be outside of the searching range. To avoid this problem, in the case that the minimized energy E_i with the depth $d_{E_{min}}$ is more than the given threshold, the depth value d is researched using Eq. (4) without limited range.

View-dependent texture mapping After deforming the mesh model, appropriate texture image for each patch is selected from the images of the original viewpoints. Concretely, for each triangle patch Ω on the 3-D mesh model, the frame number f that maximizes following function R_f is selected as the texture frame for the patch Ω.

$$R_f = \sum_{k \in \Omega} \sum_{j \in \mathbf{f}_k} \begin{cases} w_{kj}(\hat{\mathbf{p}}_k(d_{E_{min}})) & ; j = f \\ 0 & ; otherwise \end{cases}. \tag{6}$$

where $k(k \in \Omega)$ indicate a vertex number that consist of the patch Ω. $\mathbf{f}_k$ is the index list and w_{kj} is the weighting function, they were given in the deformation process. For each patch of the mesh model, image frame that maximize R_f is selected and is mapped as the texture.

Fig. 5. Example frame of input video.

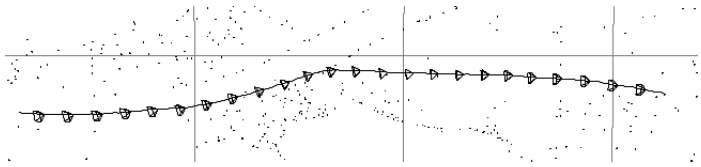


Fig. 6. Camera position and posture of OMS estimated by structure from motion. (Viewpoint is set from the sky.)

3 Experiment

In this experiment, an outdoor environment is captured using an omni-directional multi-camera system: Pointgrey research Ladybug. Ladybug has six camera units that are located to capture an omni-directional vision and each camera unit captures perspective video whose resolution is 768×1024 pixels. Fig. 5 shows an example frame of the omni-directional video (500 frames) that is captured in the target environment. We first estimate extrinsic camera parameters for this video stream [8]. Fig. 6 shows estimated camera position and posture of every 20 frames and camera path for all the frames. Point clouds in this figure indicate 3-D positions of feature points recovered in the structure-from-motion process. By using multi-baseline stereo for OMS [9], omni-directional depth map for each frame is estimated. Next, free-viewpoint images are rendered by the proposed algorithm using the parameters shown in Table 1. As illustrated in Fig. 7, in this experiment, virtual viewpoint is moved along two different routes.

(route A) the straight route in which the virtual camera goes along the original camera path.

(route B) the straight route in which the virtual camera goes away from the original camera path.

Table 1. Parameters used to generate free-viewpoint images.

Resolution of generated image	800×800[pixels]
Resolution of deformable mesh	31×31
Number of selected depth maps: N	5
Minimum number of depth maps: M	2
Rate R_d for searching-range limit	0.5

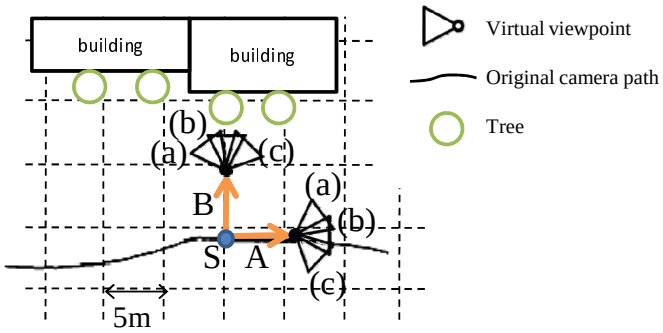


Fig. 7. Routes and directions of virtual viewpoints.

Table 2. Computational cost for free-viewpoint rendering.

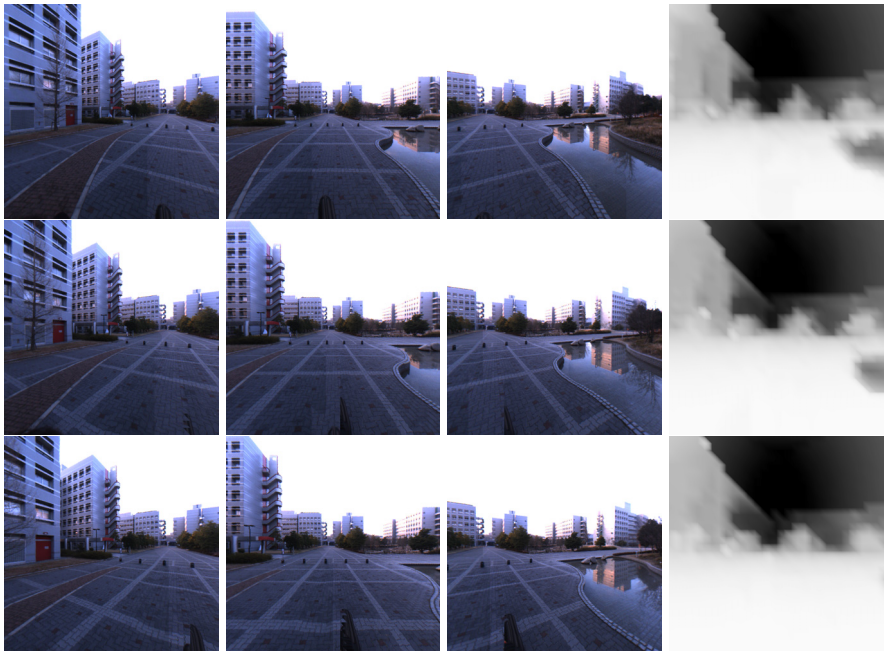
	average time [sec]
Selection of depth map	0.086
Generation of mesh model (first time of rendering)	4.190
Deformation of mesh model (except first time)	1.183
Texture mapping	5.340

For these two routes, free-viewpoint images are rendered for left (a), forward (b) and right (c) directions at every 2m distant positions from the viewpoint S.

Figs. 8 and 9 show generated images for route A and B, respectively. In these figures, depth maps that are used for the direction (b) are also shown. From these resultant images, it can be confirmed that there are no holes in the generated images. For the route A, there is very little geometric distortion in the generated images. However, there exist discontinuous textures around the center part of the generated images. That is mainly due to large frame change in texture selection for adjacent meshes. To resolve this problem, photometric correction for textures for adjacent meshes is necessary.

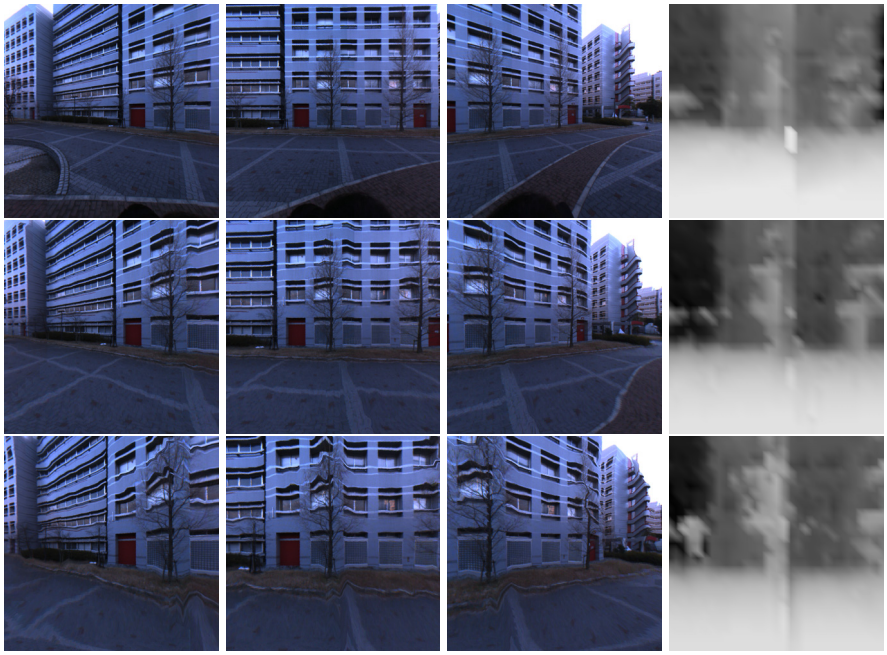
For the route B (Fig. 9), in this scene, the images were generated without large distortions if the distance from the viewpoint S to the virtual viewpoint is 2m or shorter. However, when the distance becomes 4m or more, obvious distortion can be confirmed around the tree in the scene. One of the reasons of this problem is shortage of the resolution of the mesh model. As shown in the depth maps in Fig. 9, textures on the building are mapped onto the position of the tree due to sparse depth map. In order to relax this problem without large additional cost, employment of the adaptive mesh-division will be effective.

Table 2 lists average time for each process of the proposed method in the case PC(Intel Core2Duo E8600 3.33GHz, Memory 16GB) is used. By the current implementation of the proposed method, except the initial time of the rendering stage, we need about 5.3 seconds to render single free-viewpoint image and most of time is consumed for texture mapping. In order to realize an interactive walk-



Direction (a) Direction (b) Direction (c) Depth for (b)

Fig. 8. Generated images for route A (top: 0m, mid.: 2m, bottom: 4m from point S.)



Direction (a) Direction (b) Direction (c) Depth for (b)

Fig. 9. Generated images for route B (top: 0m, mid.: 2m, bottom: 4m from point S.)

through system with the proposed method, we must optimize the current implementation by using GPU and multi-thread programming. For texture mapping process, by pre-loading the appropriate textures on texture memory, processing time will be drastically decreased.

4 Summary

In this paper, we have proposed the omni-directional free-viewpoint rendering method that uses view-dependent 3-D mesh model. In the proposed method, 3-D mesh model is deformed using the energy function that expresses the consistency of the depth maps for the original camera positions. In the experiment, free-viewpoint images are generated for several directions and positions by using omni-directional images and depth maps. In the future work, in order to improve the quality of the generated images, introduction of the adaptive mesh-division and texture blending methods are necessary.

Acknowledgments. This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (A), 19200016.

References

1. M. Goesele, N. Snavely, B. Curless, H. Hoppe and S.M. Seitz: "Multi-View Stereo for Community Photo Collections," Proc. ICCV, 2007.
2. P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.M. Frahm, R. Yang, D. Nister and M. Pollefeys: "Real-Time Visibility-Based Fusion of Depth Maps," Proc. ICCV, 2007.
3. C. Fruh and A. Zakhor: "An Automated Method for Large-Scale, Ground-Based City Model Acquisition,", IJCV, Vol. 60, No. 1, pp. 5–24, 2004.
4. B.J. King, T. Malisiewicz, C.V. Stewart and R.J. Radke: "Registration of Multiple Range Scans as a Location Recognition Problem: Hypothesis Generation, Refinement and Verification," Proc. 3DIM, pp. 180–187, 2005.
5. H.Y. Shum, S.B. Kang and S.C. Chan: "Survey of Image-based Representations and Compression Techniques," IEEE Trans. on Circuits and Systems for Video Technology, pp. 1020–1037, 2003.
6. M. Irani, T. Hassner and P. Anandan: "What Does the Scene Look Like from a Scene Point?," Proc. ECCV, Vol. 2, pp. 883–893, 2002.
7. A. Gupta, L. Goel, A. Kushal, P. Kalra and S. Banerjee : "Super Resolution of Images of 3D Scenecs," Proc. ACCV, Vol. 2, pp. 96–105, 2007.
8. T. Sato, S. Ikeda and N. Yokoya: "Extrinsic Camera Parameter Recovery from Multiple Image Sequences Captured by an Omni-directional Multi-camera System," Proc. ECCV, pp. 326–340, 2004.
9. T. Sato and N. Yokoya: "Omni-directional Multi-baseline Stereo Without Similarity Measures," Proc. OMNIVIS, pp. 193–200, 2005.