Cell-based Object Tracking Method for 3D Shape Reconstruction Using Multi-viewpoint Active Cameras

Tatsuhisa Yamaguchi, Shohei Nobuhara and Takashi Matsuyama
Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto, Japan
yamaguti@vision.kuee.kyoto-u.ac.jp, nob@vision.kuee.kyoto-u.ac.jp, tm@i.kyoto-u.ac.jp

Abstract

3D shape of objects can provide richer information for detecting, tracking or identifying the objects than a single 2D image of them. We tackle the 3D shape and texture reconstruction of an object moving in a widespread space using multi-viewpoint active cameras. Considering 3D shape and texture reconstruction, the problem in existing tracking methods using active cameras is that they cannot calibrate the active cameras accurately. We propose a cell-based tracking method that can produce multi-viewpoint images and accurate camera parameters for every frame. Our idea is to divide the space into cells and perform active camera control and calibration based on the cells. We demonstrate the performance of our method by simulation.

1. Introduction

Detecting, identifying, and analyzing the motion of objects is one of the most common research topic in computer vision and it has a wide range of applications. Several methods have been proposed so far [1]. For example, Pfinder [7] tracks and classifies human motions based on 2D blob-based statistical model.

3D features of objects, such as posture or deformation can provide more information for motion and behavior analysis. One of the approaches to obtain such 3D features is to use a parametrized 3D shape model and track objects based on it [2]. Such model-based approach works well for simple objects. However, as the complexity of objects increases, e.g. when we are trying to find a person wearing loose clothes and carrying multiple complex shaped objects, it comes difficult and ineffective to build a valid model. On the contrary, there is another type of approach that captures the 3D object shape itself first and then extract object structure by analyzing the shape. 3D video [4] is a concept of reconstructing 3D shape and texture using multi-viewpoint video. The generated data by that, a time-series 3D shape and texture of objects itself is also called '3D video' and it can provide richer information for identifying or classifying objects.

We tackle 3D shape and texture reconstruction of an object moving in a widespread area using multi-viewpoint active cameras. Several object tracking methods using active cameras have been proposed so far. For example, Ukita and Matsuyama have proposed an active camera control method for tracking moving objects [6]. It tracks multiple moving objects using multiple active cameras, but its images cannot be used for 3D video reconstruction because the cameras are not calibarted accurately. We propose a cell-based object tracking method using active cameras, that can track an object and reconstruct 3D video of it.

The requirements for 3D shape reconstruction from multi-viewpoint images can be summarized as follows:

1. Camera calibration
2. Visual coverage
3. Spatial resolution

The first requirement is the active camera calibration. 3D video reconstruction requires much more accurate camera calibration than that required for just locating object positions. However, accurate calibration of active cameras is difficult, especially when their focal lengths are changed frame by frame.

The second is visual coverage. Each point on the target object must be captured from several different viewpoints simultaneously.

The final one is the spatial resolution. The spatial resolution of final 3D video is limited by the image resolution; here we do not mean the size of each image, but the number of pixels occupied by the object in each image. In order to reconstruct high-fidelity 3D video,

objects must be captured with high enough spatial resolution.

Our idea to satisfy these requirements is to divide the space into 'cells.' We assign a set of dedicated camera control parameters that satisfy the visual coverage and spatial resolution requirements in each cell. Our tracking alrogithm restricts the active camera motion. The cameras are only allowed to 'be directed to one of the cells' using the parameter sets or switching to another cell. We discard camera images when the camera view is being switched. We calibrate the cameras for each cell before target tracking. While the active cameras are directed to a cell with single camera control parameter, they can be regarded as static cameras. It means that any camera calibration method for static cameras [8] [5] can be applied to estimate the camera parameters accurately.

2. Cell-Based 3D Video Capture Method

We propose a cell-based tracking algorithm. First we describe the overview of our algorithm. Then we formulate the problem and describe each step of our algorithm in detail.

2.1. Overview of the Algorithm

Our algorithm consists of off-line planning stage and on-line tracking stage. The off-line process generates a set of 'cells,' by dividing the target area. Then active camera control parameters for directing them to each cell, are adjusted. After that, the cameras are calibrated for each control parameters associated with each cell.

The on-line tracking process detects the object from the captured images, compute the object position by triangulation, and then controls the cameras to direct them to the cell in which the object is.

After capturing, the system generates 3D video of the object, using the captured images and active camera parameters.

2.2. Problem Formulation

Our method can be applied to the following situation:

1. There is only one target object.
2. The object movement is restricted to a given area.
3. The object's maximum velocity is given.
4. The object is not occluded by other objects for any camera.

The lowest allowable spatial resolution is specified by the user. It is represented by the minimum distance between two nearest distinguishable points on the object. We assume that the limiting factor is image resolution and do not care other optical factors such as defocusing. The spatial resolution depends on the distance of the object from each camera. In other words, each camera has limiting distance of the view that depends on the focal length and the image resolution.

We adopt the world coordinate system that has the origin at the center of the surveyed space and the z axis directed upright. We represent the object position by the projection point of its centroid to the floor and denote it by 2D coordinate (x, y). As to the object shape, we assume that the object is included by 'reference object,' a cylinder that is $2r$ in diameter, h in height and located at the same position with the object, as shown in figure 1.

The input to our algorithm consists of a scenario and resources.

Scenario

$\mathbf{D} \in \mathbb{R}^2$ Target area. A finite area within which the object moves.

$V_{\mathtt{max}}$ Maximum speed of the object.

(r, h) Size of reference object.

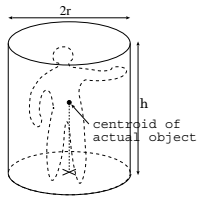


Figure 1. Reference object: a cylinder that represents the maximum allowable size of the object

Resources

$N_{\mathtt{cam}}$ Number of active cameras.

$\{O_1, O_2, \ldots, O_{N_{\mathtt{cam}}}\}$ Active camera positions.

$T_{\mathtt{s}}$ Maximum time required for changing active camera direction and focal length.

Our algorithm takes the inputs described above, and outputs a 3D video of a moving object in a widespread area. The algorithm consists of the following processes.

1. Cell generation

2. Camera grouping

3. Camera field of view(FOV) adjustment

4. Camera calibration

5. Real-time tracking

6. 3D Video generation

The following subsections describe each process in detail.

2.3. Cell Generation

We divide the target area into cells. Cells are subsets of the target area that does not intersect each other. And we also associate active camera control parameters for each cell used for directing the cameras to the cell. Using these cells and camera control parameters, we define camera control rules: When the centroid of the object is within cell j, the system directs all the cameras to cell j. We represent cells as a function that maps an object position to the cell the cameras should be directed to.

$$f_i \quad : \quad \mathbf{D} \rightarrow \{1, 2, \ldots, N_{\mathtt{cell}}\} \tag{1}$$

Here, $N_{\mathtt{cell}} \in \mathbb{N}$ is the number of generated cells in total. We also use these symbols.

$\vec{e}_i^{\,j}$ Camera control parameters for directing camera i to cell j. If the active cameras are fixed-viewpoint cameras [3], $\vec{e}_i^{\,j}$ consists of pan, tilt and zoom value.

E_i^j Camera parameters of camera i being directed to cell j, with the state specified by $\vec{e}_i^{\,j}$. Consists of intrinsic and extrinsic, geometric and photometric parameters.

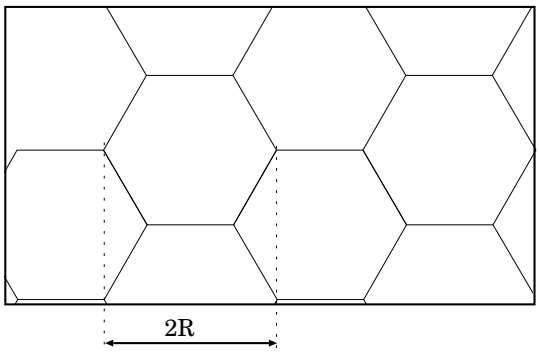


Figure 2. Hexagonal tile cell arrangement

We use hexagonal tile pattern shown in figure 2 for the shape and the arrangement of cells by the following reasons:

1. Isotropicity: We make no assumptions on the object path. So the system must be able to capture the object whichever direction the object moves to.

2. Spatial Efficiency: We can possibly adopt other isotropic tile pattern, squares or triangles. However, as mentioned later, each cell has buffer zone besides their edges. Hexagonal tile pattern minimizes the proportion of buffer zone.

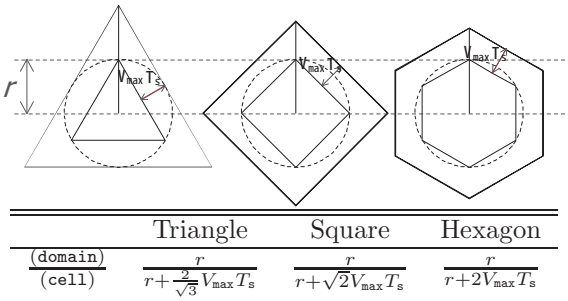


Figure 3. Proportion of cell domain to the whole cell in case of triangular, square and hexagonal cell shape, when given a common size of domain. The size of domain is approximated by a circumscribing circle.

This tile pattern has four degrees of freedom; 2-D displacement, rotation, and size of the cells. The displacement and the rotation are not so important as long as the cameras are not so close to the target area. We are currently giving these parameters manually. The size of the hexagons is critical and is discussed in section 2.3.

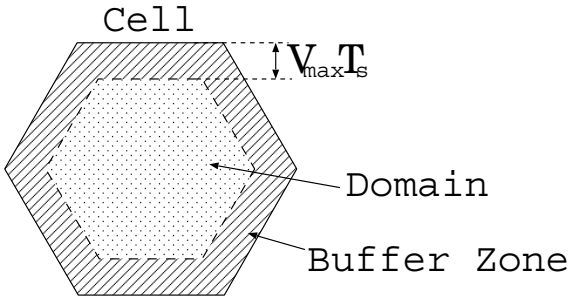


Figure 4. Buffer zone and domain of a cell

Buffer Zone and Cell Domain Switching camera views from one cell to another requires non-zero time. During this 'dead' period, the images from in-motion

cameras cannot be used for shape reconstruction because they are not calibrated. It means that when the target is close to cell borders, it cannot always be captured. We define 'buffer zone' as the part of space where the distance from the nearest cell border is shorter than $V_{\max}T_{\mathrm{s}}$. On the other hand, we name the area in cell i where the distance from any cell border is longer than $V_{\max}T_{\mathrm{s}}$ as 'domain' of cell i. When the object is in the domain of a cell, the cameras can surely capture the object with the camera control parameters associated with the cell.

The cell domains cannot cover the entire target area. In order to cover the entire target area, we divide the cameras into three groups and assign complementary arrangement of cells as shown in 5, and control each group separately. This arrangement guarantees that at least one of the camera groups is capturing the target at any time. In other words, at least a third of the cameras can contribute to reconstruct the target object shape. This is how our method satisfies the visual coverage requirement.

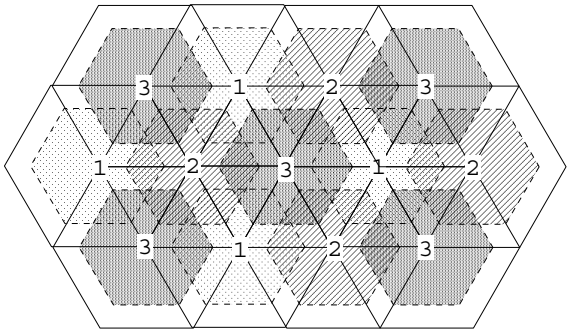


Figure 5. Complementary cell arrangement for three groups

This three-grouped cell arrangement can cover the entire space only when the size of the cells R satisfy this condition.

$$\frac{V_{\max}}{R} \leq \frac{\sqrt{3}}{6T_{\mathrm{s}}} \tag{2}$$

It means that the minimum size of cells is limited by $V_{\max}$ and T_{s}. The larger the cell is, the larger space the camera views must cover and thus the spatial resolution becomes lower. This is discussed in section 2.5 again. Therefore we adopt the minimum allowable cell size, $R = 2\sqrt{3}T_{\mathrm{s}}V_{\max}$.

2.4. Camera Grouping

The cameras in each group must be equally distributed in all the directions for reconstructing 3D video. We adopt the following scheme.

1. Sort the cameras by their azimuth angles from the center of surveyed space, and denote them as $\{A_1, A_2, \ldots, A_{N_{\mathrm{cam}}}\}$.

2. Group camera A_i into group $1 + i - 3\lfloor i/3 \rfloor$.

2.5. Camera FOV adjustment

The sufficient condition to continuously capture the object, regardless to how it moves, is that all the camera FOVs cover each domain of cell j. More precisely, every camera view in group g_l should include Minkowski sum of the domain and the reference object shown in figure 6. We denote this volume by M_j for each cell j. Our algorithm adjusts active camera control parameters in order to satisfy eq. 3. If such $\vec{e}_i^{\,j}$ does not exist, it means that the object cannot be captured with our method for the given scenario and our algorithm terminates at this step.

$$M_j \subset V_i(\vec{e}_i^{\,j}) \tag{3}$$

Here, $V_i(\vec{e}_i^{\,j})$ is the subset of the space that can be captured by camera i with control parameter $\vec{e}_i^{\,j}$.

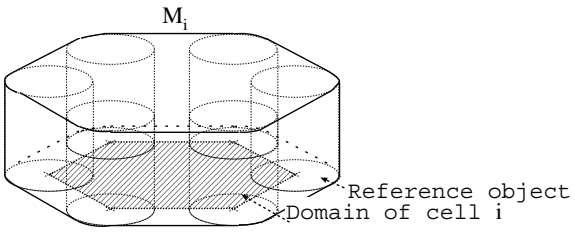


Figure 6. Minkowski sum of a domain and the reference object

Having adjusted the camera views, the spatial resolution can be computed. Let W be the number of pixels in each image row and $\phi_{i,j}$ be the horizontal angle of view with $\vec{e}_i^{\,j}$.

1. For each camera i in group l

   (a) Find the most distant point in M_j from the camera. Let its distance from the camera be $d_{i,j}$.

   (b) $s_{i,j} \leftarrow \frac{2d_{i,j}\tan\phi i,j}{W}$

2. $\max_{i,j} s_{i,j}$ is the lowest resolution

In other words, if the lowest acceptable spatial resolution were given, we can know how T_{s} and (r, h) should be set.

2.6. Camera Calibration

We calibrate the active cameras and obtain E_i^j for all $i = 1, \ldots, N_{\mathtt{cam}}, j = 1, \ldots, N_{\mathtt{cell}}$. All the active cameras can be regarded as fixed cameras, when they are directed to one of the cells. So any existing camera calibration method for fixed cameras can be applied.

2.7. Real-time tracking

The cell generation and the camera FOV adjustment process defines the camera control rule; how all the cameras should be controlled for any object position $\vec{p}$. Additionally, the camera calibration process gives a set of accurate camera parameters that enables 3D position measurement from the images. Thus the tracking can be performed by measuring the object position $\vec{p}$ from the images and controlling the active cameras in parallel.

We describe our tracking and capture processes using networked computer model. The capture process is performed by $N_{\mathtt{cam}}$ camera nodes $\pi_i (i = 1, 2, \ldots, N_{\mathtt{cam}})$, which have one camera connected each, and one master node $\pi_{\mathtt{M}}$. The nodes are connected each other to share the object position $\vec{p}(t)$. In the following descriptions, we denote the time by t and we assume that all the system clocks on the nodes are synchronized.

The measurement of the object 3D position is performed as follows. Each π_i repeats the following procedure in every time interval $\tau_{\mathtt{cam}}$.

The 2D Tracking Process on each π_i

1. Grab an image $I_i(t)$.
2. If the camera is directed to one of the cells,
   (a) Store $(t, I_i(t), j_i(t))$. Here, $j_i(t)$ is the cell number that the camera was directed to.
   (b) Track the object position on the image and compute its centroid $\vec{u}_i(t)$.
   (c) If $\vec{u}_i(t)$ is successfully computed, transmit $(t, \vec{u}_i(t), j_i(t))$ to $\pi_{\mathtt{M}}$.

The 3D Tracking Process on $\pi_{\mathtt{M}}$

1. When 2 or more sets out of $\{(t, \vec{u}_i(t), j_i(t)) \,|\, i = 1, \ldots, N_{\mathtt{cam}}\}$ have been received, compute the 3D position of the object by triangulation using $\vec{u}_i(t)$ and $E_i^{j_i(t)}$.
2. If the 3D position $\vec{p}(t)$ is successfully computed, transmit $\vec{p}(t)$ to all π_i.

And the camera control is performed as follows:

The Camera Control Process on each π_i

1. Whenever a new $\vec{p}(t)$ is received, transmit $\vec{e}_i^{f_i(\vec{p}(t))}$ to the active camera and begin switching its state.

2.8. 3D Video generation

In the algorithm described above, each π_i stores $(t, I_i(t), j_i(t))$. From these data, a sequence of multi-view images and camera parameters $\left(I_i(t), E_i^{j_i(t)}\right)$ can be obtained. It means that our method can generate a 3D video.

3. Experiment

We quantitatively evaluate our method by these two figures.

1. Camera usage: The number of the cameras contributing to reconstruct the object shape by partially or fully including the object in their views.
2. Shape fidelity: How similar the generated 3D video is to the original shape.

We simulated our algorithm and computed these figures from the tracking result.

3.1. Shape Fidelity Score

In order to quantify the visual coverage, mentioned in section 1, we define shape fidelity score for a given camera configuration and an object in a static scene. We distribute N_j sample points on the object and let $\theta_{i,j}$ be the angle of incidence of the camera i to the jth sample point, $v_{i,j}$ be the binary mask function that is 1 if jth sample point is visible from camera i. And then we define the shape fidelity score by eq. 4.

$$\frac{1}{N_{\mathtt{j}}} \Sigma_j (\max_i v_{i,j} \sin\theta_{i,j})(\max_i v_{i,j} \cos\theta_{i,j}) \qquad (4)$$

The first factor $(\max_i v_{i,j} \sin\theta_{i,j})$ represents how well the contour near point j is estimated. The second factor $(\max_i v_{i,j} \cos\theta_{i,j})$ represents how well the texture near point j can be observed.

3.2. Experiment Setup

We arranged 24 active cameras and set the target area as shown in figure 8. Other resources and scenario parameters are shown in table 1. We applied our cell generation algorithm for the scenario and resources. The cell configuration is shown in figure 9.

T_s	V_{max}	h	r
1.0[s]	300[mm/s]	1800[mm]	450[mm]

Table 1. Scenario and resources for experiment

3.3. Tracking Simulation

We implemented our algorithm for simulation. Two sequences of virtual scene were captured by our algorithm. Each virtual scene consists of a 3D video of a walking person. Figure 7 shows the model and figure 9 shows the walk paths in each sequence.

Figure 10 shows which cell the cameras were directed to at each frame. Figure 11 and 12 shows the number of cameras that contributed to shape reconstruction, for sequence 1 and 2, respectively. These figures show that the number of used cameras is exceeding 8 in most frames. The reason is that the cameras can also observe outside the cell domains despite their views being set up merely for including the cell domain. Thus they could partially contribute to reconstruct the shape of the object, by partially including the object in their images.

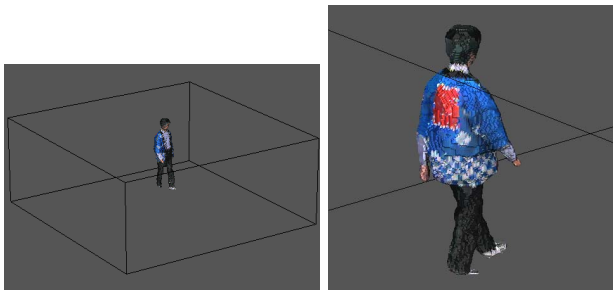


Figure 7. One of the frames in the 3D video used for simulation

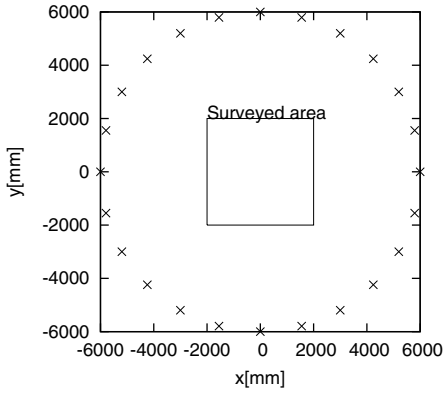


Figure 8. The camera configuration for simulation.

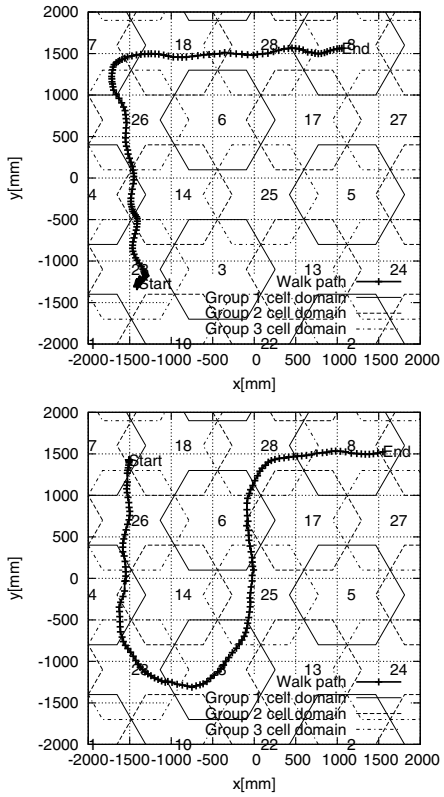


Figure 9. The object paths used for simulation and the cell arrangement. Each hexagon represents the cell domain and the numbers near the centers of them are cell ID.

Figure 13 and 14 shows one of the best and worst example of the reconstructed shape. In sequence 1, the shape fidelity score becomes the worst at frame 134. 14 shows how the reconstructed shape looks like.

4. Conclusion and Future Work

We have proposed a cell-based object tracking method, that can track an object and reconstruct 3D video of it. Our method adjusts camera views in off-line process before capturing. It enables accurate camera calibration and can judge whether the object can be captured with given resources and scenario.

One of the remaining problems is the optimization of the rules and camera views for the visual coverage and the spatial resolution mentioned in section 1. The camera control rule proposed in this paper is just a heuristic. It can only guarantee that the object is captured at least a third of the cameras, but it is not optimized for certain measure. And also, the cameras in one group is controlled with the same rule. Since the geometric property of the cameras are not uniform, we predict that controlling each camera separately could

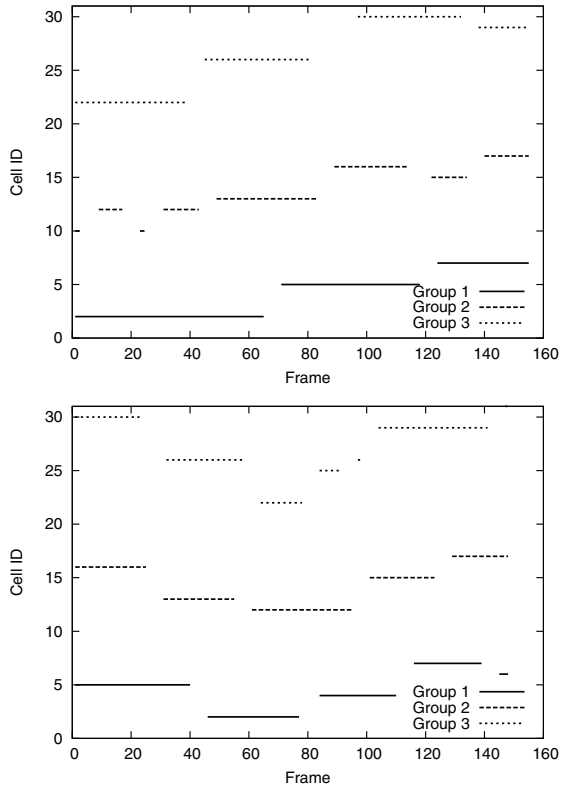


Figure 10. The timing chart of the active camera control for each sequence. Vertical axis represents the cell that the cameras in each group were directed to. Blank part indicates that the cameras were in motion.

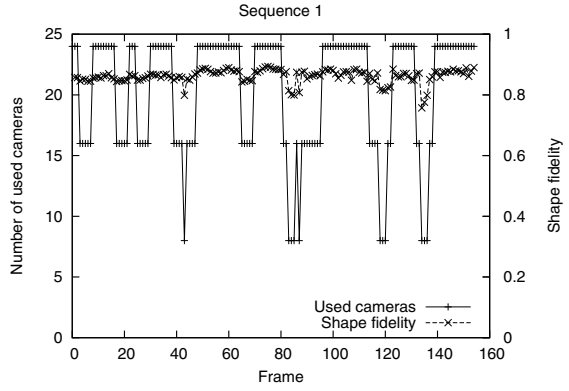


Figure 11. Number of the cameras contributed to shape reconstruction, with sequence 1

improve the efficiency of camera usage. As mentioned in section 2, f_i is a map from a plane to integer. It means that we would need a good parametrization and a good optimization scheme to find the best one instance of the cell arrangement. This is what we would address as the next step.

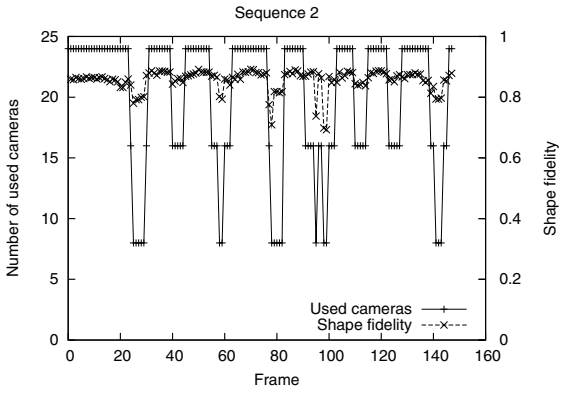
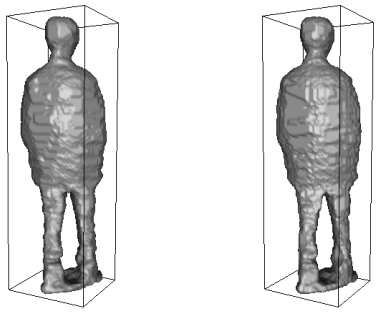


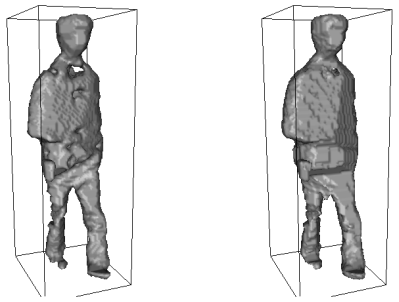
Figure 12. Number of the cameras contributed to shape Reconstruction, with sequence 2



Original

Reconstructed
Shape fidelity = 0.857
24 cameras

Figure 13. Original and reconstructed shape of frame 1 from sequence 1.



Original

Reconstructed
Shape fidelity = 0.757
8 cameras

Figure 14. Original and reconstructed shape of frame 134 from sequence 1.

Acknowledgements

This research was supported by "Foundation of Technology Supporting the Creation of Digital Media

Contents" project (CREST, JST) and Ministry of Education, Culture, Sports, Science and Technology under the Leading Project: "Development of High Fidelity Digitization Software for Large-Scale and Intangible Cultural Assets".

References

[1] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(3):334–352, Aug. 2004.

[2] I. A. Karaulova, P. M. Hall, and A. D. Marshall. A hierarchical model of dynamics for tracking people with a single video camera. In *British Machine Vision Conference(BMVC)*, pages 262–352, 2000.

[3] J. Kondou, X. Wu, and T. Matsuyama. Calibration of partially-fixed viewpoint active camera. *IPSJ SIG Notes. CVIM (in Japanese)*, 2003(36)(137-19):149–156, 20030327.

[4] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real–time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualization for 3D video. *Computer Vision and Image Understanding*, 96(3):393–434, 2004.

[5] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence: Teleoper. Virtual Environ.*, 14(4):407–422, 2005.

[6] N. Ukita and T. Matsuyama. Real-time cooperative multi-target tracking by communicating active vision-agents. *Computer Vision and Image Understanding*, 97(2):137–179, 2005.

[7] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, Jul 1997.

[8] Z. Zhang. A flexible new technique for camera calibration. *IEEE transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.