社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－CVIM－157 (28)
2007／1／12

空間分割による広範囲実時間人体の３次元形状復元

ウ 小 軍 † 大澤達哉 † 数藤恭子 †
若林佳織 † 安野貴之†

近年、動く対象の実時間３次元形状復元手法として、視体積交差法が注目を浴びている。いくつかの高速化手法も提案されている。しかし、原理的に形状復元に用いる多視点カメラ群の全ての画像内に対象が完全に撮影されるという制約がある。この制約のため、広範囲に動く人物の３次元形状復元が困難となる。本発表では、こういった撮影時あるいはカメラ配置上の制約を無くすため、部分的に撮影された対象画像からも完全な３次元形状を復元する手法を提案する。これによって、原理的に形状復元の範囲を広げることが可能となった。さらに、復元範囲の拡大による計算量の増加を押えるため、全体の復元空間を複数の復元領域に分割し、対象の存在し得る領域についてのみ形状復元を行う手法を提案する。実験によって、提案手法を有効性を示す。

Wide Area Real-Time 3D Shape Reconstruction from Partially Observed Silhouettes

Xiaojun Wu ,† Tatsuya Osawa ,† Kyoko Sudo ,†
Kaoru Wakabayashi † and Takayuki Yasuno†

Recently, several methods based on the theory of shape from silhouettes have been developed to realize real-time 3D rendering of free viewing-point videos. Due to the limitation that complete silhouette of the target must be observed from every viewing-points, in such systems, the target, mostly a person, have to be en-caged within a narrow area or only one part of the target is reconstructed. In this paper, we propose an algorithm of 3D shape reconstruction from partially observed silhouettes at first. By this extension, full 3D shape can be reconstructed over a wide area, so that it becomes possible to obtain the full 3D shape of natural motions. Such 3D motion models can be utilized for 3D rendering , motion analysis, action understanding and so on. Meanwhile, the computation cost increases when the reconstruction area becomes wider. To keep efficient computation, we adopt the octree searching algorithm to detect the target area in the wide 3D space with tiny overhead. By taking the target area as the computation region, it is possible to develop a real-time 3D shape reconstruction system. Experimental results are shown to prove the efficiency and effectiveness of our proposed algorithms.

1. Introduction

Nowadays, the shape from silhouettes (SFS) method[1] or volume intersection method(VIM) has been well known to obtain the shape of a dynamic target, such as a moving human being. Such dynamic 3D shape data can be widely utilized for the analysis of human motions, high-reality human interface systems (VR, or MR), high fidelity 3D video contents and so on[2)~10)]. The voxel representation of the 3D shape is popular in such researches. For example, in[11)], the accelerated plane-to-plane projection algorithm have been proposed to realize real-time volume. They have also parallelized the algorithm and have it implemented on a PC cluster[12)]. By using pipeline processing model, near video rate (30fps) of full 3D shape reconstruction has been achieved by their system[13)]. Also W. Matusik et la has proposed the algorithm for directly computing the 3D shape as polyhedral visual hull[14)]. Since all of these methods are based on the concept of SFS, the target must be observed from all viewing points. Due to this limitation, the reconstruction region is set as a cube of $1 \times 1 \times 2\mathrm{m}^3$ in[13)], which is obviously too narrow for a person to act naturally. For the same reason, in[15)] only the upper half part of the body is taken as the

† NTT サイバースペース研究所
NTT Cyber Space Laboratories

target. Furthermore, the reconstruction region becomes much more narrow while adding cameras to such systems.

To extend the area for 3D shape reconstruction, there exist the following two issues.

(1) Camera layout problem
From the viewpoint of efficiency of the multiple camera systems for a wide area, it is better to have each camera viewing different areas. That is, only one part of the whole area is captured by one of the cameras. To reconstruct the full 3D shape of the target with such camera layout, the problem of how to reconstruct the full shape from partial observed images (silhouettes) must be resolved.
By applying active tracking system with active cameras, i.e. the pan-tilt cameras for example, we can have the common viewing area of the cameras changing dynamically. But it is hard to have each camera capturing the perfect target at each time. To reconstruct the full 3D shape with such active camera system, it also need to resolve the above problem of shape from partial silhouettes.

(2) Computational complexity problem
When the area is extended, the number of voxels in the space increases greatly. It is obviously inefficient to compute all of the voxels each time. The problem of computational complexity for wide area shape reconstruction must also be resolved.

In this paper, for the first issue, we propose the algorithm for computing the full body 3D shape from multiple partially observed silhouettes. By this extension, there is no need for all the cameras to capture the perfect silhouettes and it becomes possible to obtain the full body shape while the target moving around a wide area. For the second issue, we adopt the octree searching algorithm to determine the computation region before a fine reconstruction. In what follows, we will firstly show the extended algorithm of 3D shape from partially observed silhouettes in section 2. After that, we will introduce the coarse-to-fine reconstruction method in section 3. Finally, several experimental results are shown in section 4 to prove both the effectiveness and efficiency of our proposed methods.

2. 3D Shape from Partially Observed Silhouettes

At first, we will show an "imperfect" 3D shape sample in Fig. 1, where the head part of the person is excluded. The imperfect shape is computed by the naive SFS algorithm. Since not all of the cameras are configured to capture the full body of the person perfectly, the head part is excluded. This sample of the imperfect 3D shape shows the limitation of the naive SFS algorithm. In this section, we will show our extended SFS algorithm to get rid of such limitation. Before showing the details, the naive SFS algorithm is summarised and some notations are defined at first.

2.1 Naive Shape from Silhouettes Algorithm

Fig.2(a) shows one frame of the silhouette observed by one of the cameras. The black means the background and the white means the target. In the figure, the gray polygon means the projected voxel on the screen. Let i be the camera index of the total N cameras system. Let v denote one voxel. For each voxel v, let $w(v)$ denote the occupation state of v. That is, v is occupied by the target if $w(v) = 1$, or is empty if $w(v) = 0$. $w(v)$ is simply computed as the following equation.

$$w(v) = \prod_{N} w_i(v) \tag{1}$$

where $w_i(v)$ is defined for each camera as below.

$$w_i(v) = \begin{cases} 1 & (v \text{ is projected on the target,} \\ & \text{as shown in Fig.2(b) or (c))} \\ 0 & (\text{otherwise,} \\ & \text{as shown in Fig.2(a))} \end{cases} \tag{2}$$

Notice that if v is projected out of the picture, $w_i(v)$ is defined as 0, and the voxel will be calculated as empty by Equation (1).

2.2 Extension Algorithm Using Partially Observed Silhouettes

It is clear that the limitation of the naive SFS is caused by the definition in Equation(2), where the state that the voxel can not be observed by the camera is not distinguished from the state that the voxel is projected on the background. Therefore, in our algorithm, we introduce the representation of such case of "outside of viewing-field" as shown in Fig. 3.

Taking "outside" part into the definition, there exist total 6 different cases when projecting one voxel to one camera screen. 3 of the cases are shown

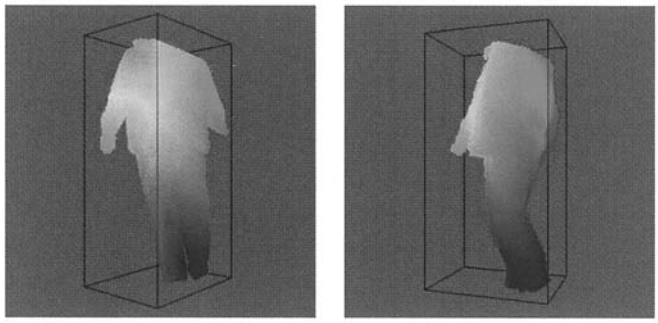

Fig. 1 Reconstruction Sample Using Naive Volume Intersection: due to the target is not perfectly observed from all of the cameras, the head part is excluded in the reconstructed 3D shape.

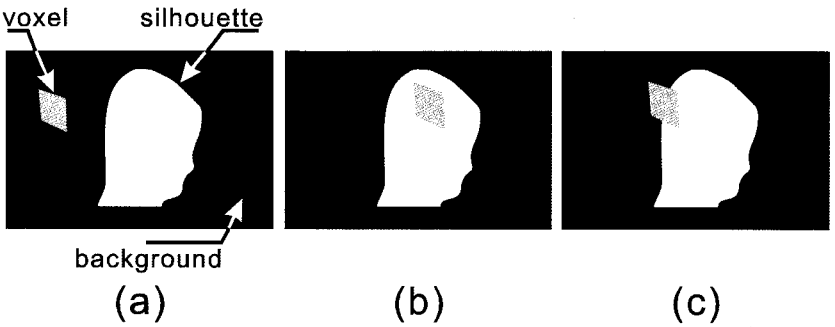

Fig. 2 The naive volume intersection algorithm

in Fig.4 and other 3 cases are shown in Fig.5. In addition to the variable $w_i(v)$ for each voxel v projected on i-th camera, $o_i(v)$ is introduced to represent whether v is observed outside of the viewing-field or not. $o_i(v)$ is defined by the following equation.

$$o_i(v) = \begin{cases} 1 & (v \text{ is projected outside,} \\ & \text{i.e. cases shown in Fig.4)} \\ 0 & (\text{otherwise,} \\ & \text{i.e. cases shown in Fig.5)} \end{cases} \quad (3)$$

Here, the definition of $w_i(v)$ is modified as follows.

$$w_i(v) = \begin{cases} 1 & (v \text{ is projected on target,} \\ & \text{i.e. cases shown in} \\ & \text{Fig.4(a-3) or Fig.5(b-2),(b-3))} \\ 0 & (\text{otherwise,} \\ & \text{i.e. cases shown in} \\ & \text{Fig.4(a-1), (a-2) or Fig.5(b-1))} \end{cases} \quad (4)$$

According to the definition, for the voxel v, if $o_i(v) = 1$ it means v is not entirely observed by i-th camera. For the N cameras system, if $\sum_N o_i(v) = N$ it means v is not entirely observed by any of the cameras. That is, the occupation of the voxel can hardly be determined by the system. On the other hand, if $\sum_N o_i(v) = 0$ it means the voxel is observed by all cameras and the occupation can be determined by the naive SFS shown above. So we take the number of $\sum_N o_i(v)$ as the reliability of the system for voxel v, and a threshold $T(v)$ is introduced for voxel v, and the occupation computing is conducted as what follows.

At first, project the voxel v to all N cameras, we then have the set of $\{w_i(v)|i = 1, 2, \ldots, N\}$ and the set of $\{o_i(v)|i = 1, 2, \ldots, N\}$. Here we define the set $W(v)$ as below.

$$W(v) = \{w_i(v)|\text{if} o_i(v) = 0, i = 1, 2, \ldots, N\} \quad (5)$$

Secondary, we define a reliable intersection function $w'(v)$ as below.

$$w'(v) = \prod_{w_i(v) \in W(v)} w_i(v) \quad (6)$$

$w'(v)$ is just the occupation result calculated by the cameras where v is entirely observed. By introduc-

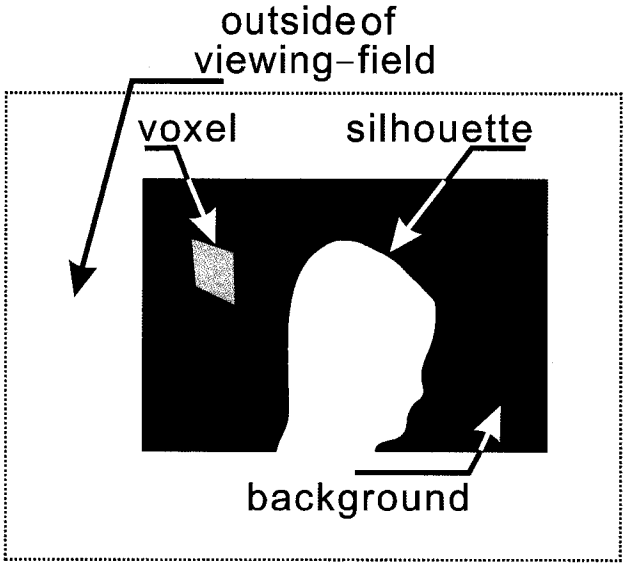

Fig. 3 In addition to "target" and "background", we also deal with the place of "outside of viewing-field"

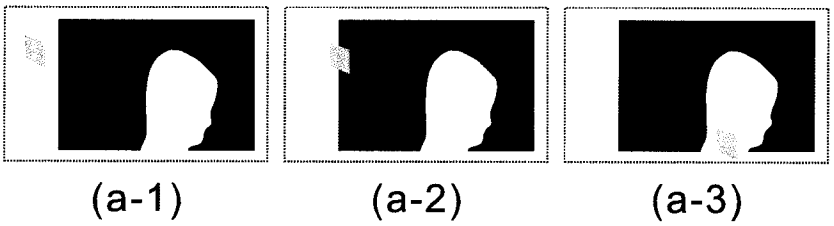

Fig. 4 Cases of the voxel is projected outside.

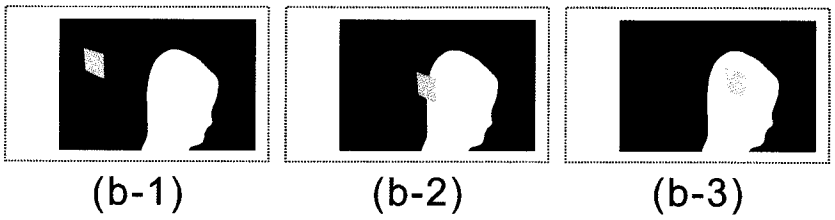

Fig. 5 Cases of the voxel is projected inside

ing a threshold $T(v)$, the final occupation computation can be done by the following equation.

$$w(v) = \begin{cases} w'(v) & \sum_N o_i(v) < T(v) \\ \prod_N w_i(v) & \text{otherwise} \end{cases} \quad (7)$$

Notice that the threshold is defined as a function of v. It means the reliability of the system is location-oriented. In practice, the reliability is determined by the camera layout. That means, if the cameras are fixed, for each voxel, the number of cameras which the voxel is entirely observed can be calculated before hand and the threshold can be determined from that number.

By the above extension, for each voxel, the input cameras are filtered by its viewing-field. As a result, there is no need for all cameras to entirely observe all voxels, of which the computation region is composed. So the full complete 3D shape can be calculated from partially observed silhouettes.

3. Fast Target Area Detection Using Octree Searching Algorithm

Now, the computation region of the shape reconstruction can be enlarged by applying the shape from partial silhouettes algorithm. Generally, the computational complexity increases when the computation region becomes wide. This is also true for the system of parallel computing system as shown in[13]. That is the computation time is longer in proportion to the reconstruction region. To achieve the real-time shape reconstruction over a wide area, we propose a coarse-to-fine reconstruction to increase the computational efficiency. The details are shown as follows.

In practice, we define the target area as a set of multiple cubes (or large size voxels). For the real-time processing, the detection of the target area should be conducted at high speed. So the octree searching algorithm is adopted for this task. The processing flow is shown in Fig.6, which is well known and the details of the algorithm is omitted. Moreover, the multi-resolution input silhouette images pyramid is also prepared and the image with proper resolution is selected according to the voxel size during the projection computing. Therefore, the target detection is accelerated much more.

Since the target area is defined as a set of multiple cubes, the result cubes cover the target in the 3D space. Since the reconstruction region is defined as a single cube in[13], it is obvious both easy and practical to extend the region as multiple cubes. Furthermore, by embedding the target detection task as one of the pipeline stages, the total throughput will not be damaged in a pipeline processing system shown in[13]. The performance of octree searching task is evaluated in the next section.

4. Performance Evaluation

4.1 Entire 3D Shape Reconstructed from Partially Observed Silhouettes

At first, the shape computed using our extended SFS algorithm is shown in Fig. 7. The input silhouettes are same as the reconstruction of Fig. 1. This shows extension is effective to for the reconstruction from partial observed silhouettes.

Secondly, we evaluate the algorithm using multi-view sequences of a moving person. Fig.8 shows the samples of input silhouettes of a moving person observed by 3 of the total 16 cameras of our system. It is clear that the target is not entirely observed by all the cameras. Fig.9 shows one frame of 3D shape reconstructed, rendered in 6 different viewing directions. The entire 3D shape can be confirmed. The computation region is $3 \times 3 \times 3\mathrm{m}^3$, which is also the moving area for the target.

4.2 Fast Target Detection

Capturing from 16 cameras, the target detection performance is measured in this experiment. The reconstruction space is as same as the above experiment, i.e. $3 \times 3 \times 3\mathrm{m}^3$, and the minimal voxel size for representing target area is 37.5cm. The CPU is a Xeon of 3GHz. The performance is evaluated for the multi-viewpoint sequences of 100 frames. Both the detecting time and the volume of the target area (i.e. the sum of the voxels' volume) are plotted in Fig.10. The horizontal axis is the frame index. The left vertical axis is the measurement of the target area's volume, which is plotted as blue diamond dots. For comparison, the constant volume of the reconstruction region set in[13] is also shown as a red line in the graph. The right vertical axis is the measurement of the detecting time, which is plotted as black circle dots.

From the graph, the detecting rate is faster than about 30fps and is fast enough for embedding the target detection to the pipeline processing. Furthermore, the detected area is much smaller than the fixed reconstruction region set in[13]. That means the efficiency of the shape reconstruction can also be improved by introducing the target area detection phase.

5. Conclusion

In this paper, we describe the extended algorithm for 3D shape reconstruction from partially observed silhouettes. By this extension, we can setup a multi-viewpoint cameras system for the 3D shape reconstruction in a wide area. For the real-time computation, we developed the fast target area detection method using octree search algorithm. The experimental results shows both the effectiveness of the extended algorithm and the efficiency of the detection method. In addition, since the overhead of these extensions is small enough, the real-time 3D shape reconstruction in a wide area can be realized by applying the extensions on the PC cluster system. In fact, we are implementing our proposed algorithms on the real-time parallel volume intersection system and we plan to report the perfor-

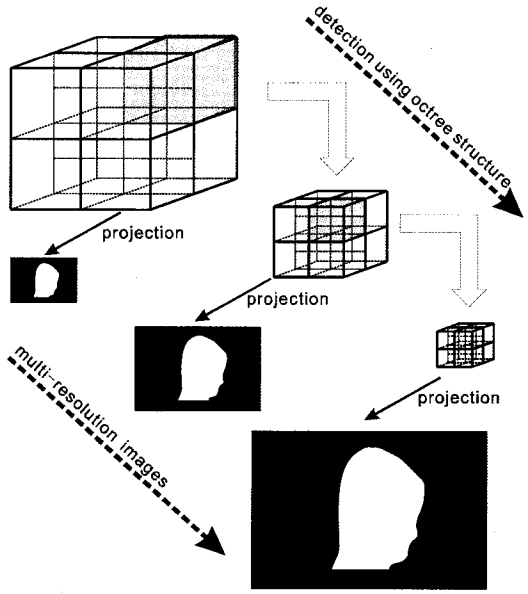

Fig. 6 Fast target area detection

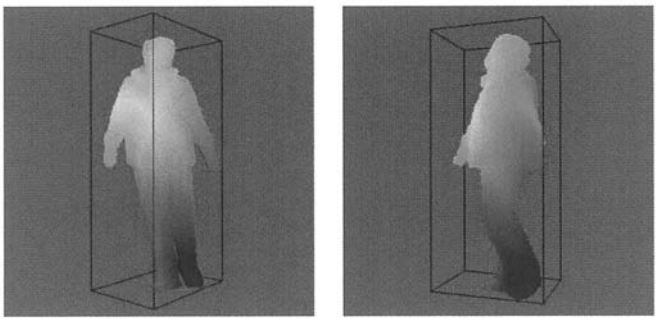

Fig. 7 Reconstruction Sample Using Extended Shape from Silhouette: although the target is not perfectly observed from all of the cameras, the full body is reconstructed perfectly.

mance of such online system in a near future.

References

1) Laurentini, A.: How far 3d shapes can be understood from 2d silhouettes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17(2), pp. 188–195 (1995).
2) Seitz, S. M. and Dyer, C. R.: Toward Image-Based Scene Representation Using View Morphing, *Proc. 13th Int. Conf. on Pattern Recognition, Vol. I*, pp. 84–89 (1996).
3) Matusik, W., Buehler, C., Raskar, R., Gortler, S. J. and McMillan, L.: Image-based visual hulls, *Proc. of SIGGRAPH 2000*, ACM Press/Addison-Wesley Publishing Co., pp.369–374 (2000).
4) Kitahara, I., Ohta, Y. and Kanade, T.: 3D Video Display of Sports Scene using Multiple Video Cameras, *Meeting on Image Recognition and Understanding, vol 1*, pp. 3–8 (2000).
5) Sugawara, S., Suzuki, G.and Nagashima, Y., Matsuura, M., Tanigawa, H. and Moiriuchi, M.: InterSpace: Networked Virtual World for Visual Communication, pp. 1344–1349 (1994).
6) Moezzi, S., Tai, L. and Gerard, P.: Virtual View Generation for 3D Digital Video, *IEEE Multimedia*, pp. 18–26 (1997).
7) Kanade, T., Rander, P. and Narayanan,

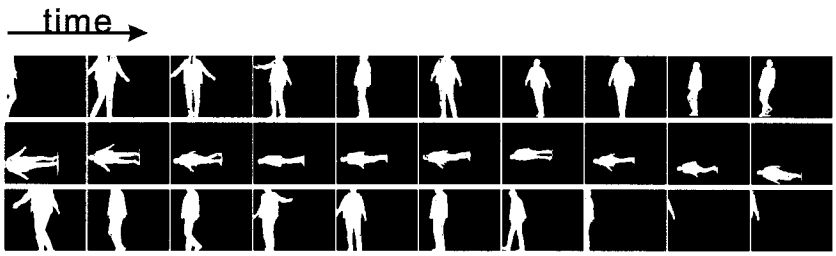

Fig. 8 Sihouettes sequence samples

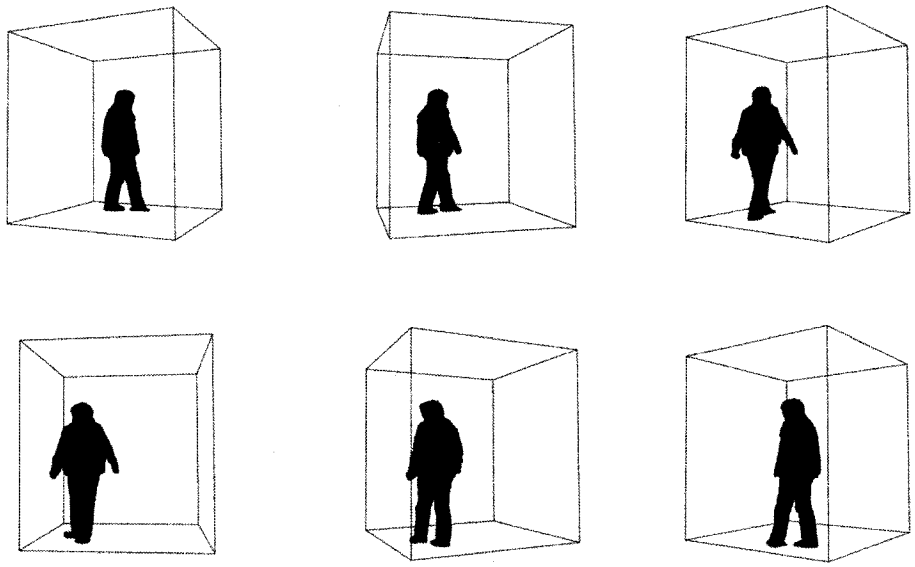

Fig. 9 One Frame of Entire Shape Reconstructed from Partially Observed Silhouettes

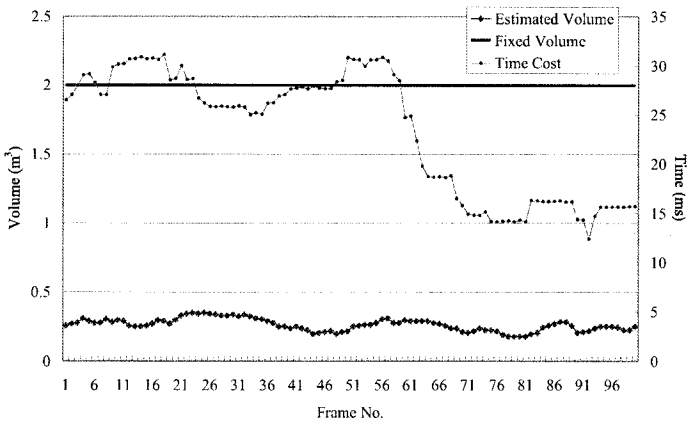

Fig. 10 Performance evaluation

P. J.: Virtualized Reality: Constructing Virtual Worlds from Real Scenes, *IEEE Multimedia*, pp. 34–47 (1997).
8) Borovikov, E. and Davis, L.: A Distributed System for Real-Time Volume Reconstruction, *Proc. of International Workshop on Computer Architectures for Machine Perception*, Padova, Italy, pp. 183–189 (2000).
9) Cheung, G. and Kanade, T.: A Real Time System for Robust 3D Voxel Reconstruction of Human Motions, *Proc. of Computer Vision and Pattern Recognition*, South Carolina, USA, pp. 714–720 (2000).
10) Vedula, S.: *Image Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events*, PhD Thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2001).
11) Wada, T., Wu, X., Tokai, S. and Matsuyama, T.: Homography Based Parallel Volume Intersection: Toward Real-Time Reconstruction Using Active Camera, *Proc. of International Workshop on Computer Architectures for Machine Perception*, Padova, Italy, pp. 331–339 (2000).
12) Matsuyama, T., Wu, X., Takai, T. and Wada, T.: Real-Time Dynamic 3D Object Shape Reconstruction and High-Fidelity Texture Mapping for 3D Video, *IEEE Trans. on Circuit and Systems for Video Technology*, Vol. 14, pp. 357–369 (2004).
13) Wu, X., Takizawa, O. and Matsuyama, T.: Parallel pipeline volume intersection for real-time 3D shape reconstruction on a PC cluster, *Proc. of The 4th IEEE International Conference on Computer Vision Systems*, New York, USA (2006).
14) Matusik, W., Buehler, C. and McMillan, L.: Polyhedral Visual Hulls for Real-Time Rendering, *Procóf the 12th Eurographics Workshop on Rendering Techniques*, pp. 115–126 (2001).
15) Baker, H., Tanguay, D., Sobel, I., Gelb, D., Gross, M., Culbertson, W. and Malzbender, T.: The coliseum immersive teleconferencing system, *Proc. of International Workshop on Immersive Telepresence*, Juan Les Pins, France (2002).