Real-time Free-viewpoint Video Generation Using Multiple Cameras and a PC-cluster

Ueda, Megumu
Department of Intelligent Systems, Kyushu University

Arita, Daisaku
Department of Intelligent Systems, Kyushu University

Taniguchi, Rin-ichiro
Department of Intelligent Systems, Kyushu University

出版情報：Proceedings of International Conference on Computer Graphics and Imaging, pp.87-92, 2004-08
バージョン：
権利関係：

Real-Time Free-Viewpoint Video Generation Using Multiple Cameras and a PC-Cluster

Megumu Ueda, Daisaku Arita, and Rin-ichiro Taniguchi

Department of Intelligent Systems, Kyushu University
6-1, Kasuga-koen, Kasuga, Fukuoka 816-8580 Japan
TEL:+81-92-583-7618, FAX:+81-92-583-1338
{ueda, arita, rin}@limu.is.kyushu-u.ac.jp

Abstract. In this paper, we propose a system generating free-viewpoint video using multiple cameras and a PC-cluster in real-time. Our system firstly reconstructs a shape model of objects by the visual cone intersection method, secondly transforms the shape model represented in terms of a voxel form into a triangular patch form, thirdly colors vertexes of triangular patches, lastly displays the shape-color model from the virtual viewpoint directed by a user. We describe details of our system and show some experimental results.

Keywords: Image-based rendering, Shape and color reconstruction, Real-time computer vision, Parallel computer vision

1 Introduction

Currently, televisions are used for real-time, or live distribution of scenes in the world. In television, however, a video captured by a camera are displayed on a screen and the viewpoint is chosen only among camera positions, not among arbitrary positions. On the other hand, computer graphics techniques can generate a free-viewpoint video, in which a viewer can changed the viewpoint to arbitrary positions. However, computer graphics require a structure and motion model of objects and it is time consuming to construct such a model in advance. Then, we aim to construct a computer graphics model by computer vision techniques in real-time for generating live free-viewpoint videos.

Several researches have been done for generating free-viewpoint videos using multiple cameras since Kanade et al.[1] had proposed the concept of "Virtualized Reality". We can classify such researches into two approaches. The first approach reconstructs 3D shapes of objects and the second one does not reconstruct them. We select the first approach because it has more applications such as motion analysis, reflectance analysis and so on. As the first approach, Matsuyama et al.[2] and Carranza et al.[3] have developed systems which generate a computer graphics model from multiple camera videos. However, they cannot generate the model in real-time since precise shape reconstruction and model coloring from multiple images consume a lot of time. In comparison with these systems, our system cannot reconstruct a shape model precisely. However, our system can generate free-viewpoint video in real-time because of a new model coloring method proposed in this paper. Our system can be used for live videos from arbitrary viewpoints.

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3331, pp. 418–425, 2004.

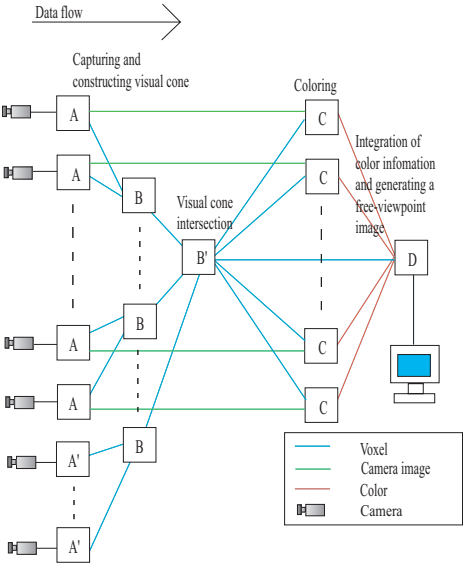


Fig. 1. System configuration

2 Free-Viewpoint Video Construction

Our system realizes real-time free-viewpoint video generation using a PC-cluster and RPV[4] which is a programming environment for real-time image processing on a distributed parallel computer such as a PC-cluster. Multiple cameras synchronized by an external trigger are placed in a convergent setup around the center of the scene. Each camera is connected to a PC.

The processes for generating free-viewpoint videos in real-time are as follows.

1. Reconstructing a shape model of objects by the visual cone intersection method[5].
2. Transforming the shape model represented in terms of a voxel form into a triangular patch form by the discrete marching cubes method[6].
3. Coloring vertexes of triangular patches varying with the position relation between the virtual viewpoint directed by a user and the viewpoints of cameras.
4. Displaying the shape-color model from the virtual viewpoint with painting triangular patches by interpolating among vertexes.

These processes are distributed to PCs shown in Fig. 1 and executed in pipeline parallel.

2.1 Node-A

First, each node-A extracts object silhouettes from video frames captured by a camera by background subtraction and noise reduction. Secondly, each node-A constructs visual cones. A visual cone is defined as a cone whose apex is the viewpoint and whose cross

section coincides with the silhouette of the object. Visual cones are represented in terms of a voxel space. Lastly, each node-A sends the visual cones to a node-B and sends the colored silhouette image to a node-C.

Each node-A' has same functions as a node-A without sending the colored silhouette image. This means that a node-A' does not works for model coloring but only for shape model reconstruction. Shape model reconstruction requires as many cameras as possible. On the other hand, model coloring needs large processing time, and then it is difficult to use many cameras. The number of node-A's is determined based on the balance between model coloring precision and processing power.

2.2 Node-B

Each node-B gathers and intersects visual cones from multiple viewpoints to construct a shape model of the objects represented in terms of a voxel space. Since this process is time consuming, it is distributed to multiple node-Bs hierarchically. Node-B', the last node of node-Bs, transforms the finale shape model represented in terms of a voxel space into that in terms of triangular patches by the discrete marching cubes method. However, node-B' sends the voxel space and its corresponding patterns of the discrete marching cubes method instead of triangular patches since the triangular patch form is not efficient from the viewpoint of data size.

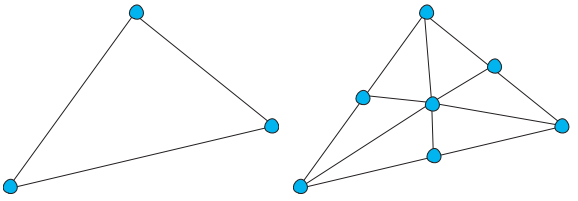
2.3 Node-C

First each node-C transforms the shape model represented in terms of a voxel space into those of triangular patches by the discrete marching cubes method using patterns sent from node-B'. Secondly, each node-C colors visible vertexes of the shape model based on one camera image. At this time, each triangular patch is divided into six triangular patches as shown in Fig. 2 since increasing the number of vertexes makes coloring resolution higher without lengthening processing time for shape reconstruction. Lastly, each node-C sends color information of all vertexes of the shape model.

For coloring vertexes in real-time, it is necessary to quickly judge whether each vertex is visible from the camera or not. Conservative visibility check method has to check whether each vertex is occluded by each triangular patch. That computation amount is $O(N^2)$, where N is the number of vertexes. So we propose a new method based on the Z-buffer method, whose computation amount is $O(N)$. Our method consists of two steps (See Fig. 3). At the first step, node-C searches for the object surface which faces against the viewpoint and which is nearest to the viewpoint in each pixel p. This step is realize by the Z-buffer method altered to taking account of not all surfaces but only surfaces facing against the viewpoint. Then, node-C lets d_p be the distance between the viewpoint and the nearest surface. At the second step, node-C colors all vertexes which faces toward the viewpoint and which is nearer to the viewpoint than d_p in each pixel p. The color of the vertexes is that of pixel p.

2.4 Node-D

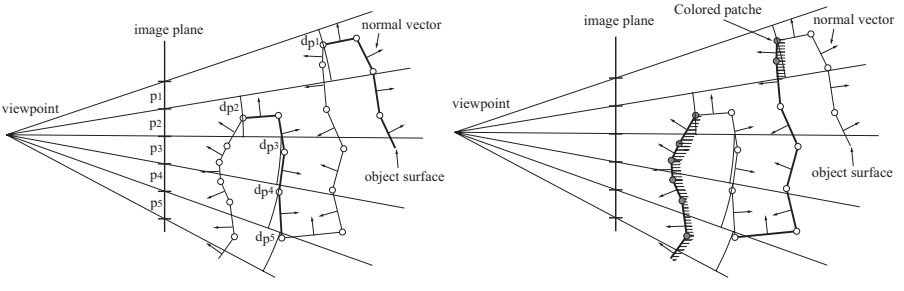
First node-D receives the position of the virtual viewpoint directed by a user.



(a) Original triangular patch

(b) Divided triangular patch

Fig. 2. Dividing triangular patch



(a) step 1

(b) step 2

Fig. 3. Coloring vertexes

Secondly node-D transforms the shape model represented in terms of a voxel space into those of triangular patches by the discrete marching cube method in the same way as node-C. There are two reasons why shape model transformation is made on both node-C and node-D. The first one is because the data size of triangular patches is very large and the time to transport triangular patches is too long. The second one is because processing times of node-B, node-C and node-D are balanced best.

Thirdly, node-D integrates color information of all cameras. The integrated color value for each vertex is weighted mean of color value from node-C. The weight W_n of camera n is calculated by the following expression;

$$W_n = \frac{(cos\theta_n + 1)^\alpha}{\sum_{x=0}^{N}(cos\theta_x + 1)^\alpha} \tag{1}$$

where N is the number of cameras visible the vertex, θ_n is the angle between the vector from the virtual viewpoint to the vertex and that from the camera viewpoint to the vertex

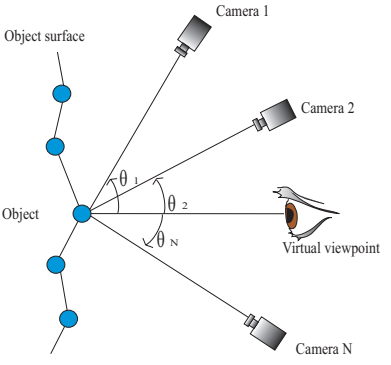


Fig. 4. Angle between camera and virtual viewpoint

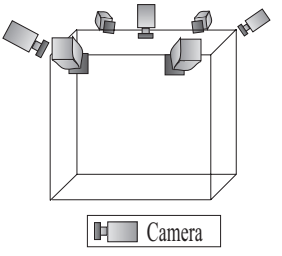


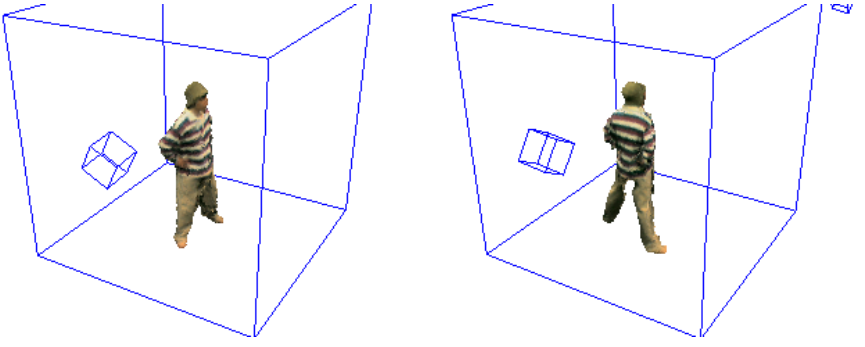
Fig. 5. Camera arrangement

(a) viewpoint 1 (b) viewpoint 2

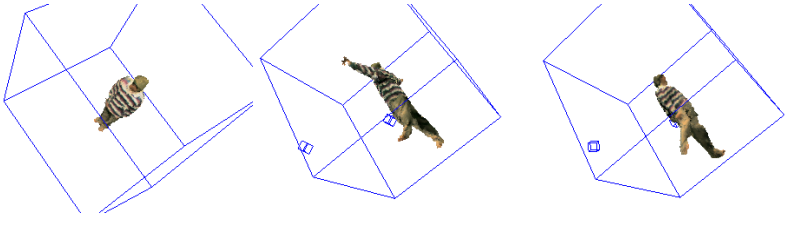
Fig. 6. Camera images(upper) and generated images(lower)

(See Fig 4). In this experiment, we let α be 1. Color value of a vertex visible from no camera is let be same as the mean value of neighbor vertexes.

Lastly, node-D generates an image from the directed viewpoint. Each triangular patch is painted by interpolating among vertexes acceleratedly on a state-of-the-art consumer-grade graphics card.



(a) virtual viewpoint A (b) virtual viewpoint B



(c) virtual viewpoint C (d) virtual viewpoint D (e) virtual viewpoint E

Fig. 7. Generated virtual-viewpoint images

3 Experiments

Using our proposed system, we generate free-viewpoint video in real-time to evaluate the precision of generated images, processing time of each node, latency, and the amount of data transfer. We use 17 PCs (six node-As, one node-A', two node-Bs, one node-B', six node-Cs, one node-D), each of which has an Intel Pentium4 (3GHz), 1GB memory and NVidia GeForce FX. PCs are connected by Myrinet, which is a kind of Gigabit network. And we use seven IEEE1394-based digital cameras, whose resolution is 320×240, arranged as shown in Fig. 5. The ceiling camera is connected to the node-A'. All cameras are calibrated in advance by Tsai's method[7]. Voxel space resolution is $128 \times 128 \times 128$ and the size of a voxel is 2cm.

Fig. 6 shows two pairs of a camera image and a generated image whose viewpoints are same. And Fig. 7 shows five generated images from virtual viewpoints. Each image is well-generated.

Fig. 8 shows the sum of root mean square errors between a camera image and a generated image with a same viewpoint. This may be caused by

- shape reconstruction error,
- camera calibration error,
- re-sampling error from image pixels to triangular patch vertexes, and
- color integration error.

Table 1. Amount of data sending from each node

Node	Average (Kbyte)
A (Image)	93.5 (variable)
A and B (voxel)	256.0 (constant)
B'	28.6 (variable)
C	92.0 (variable)

Fig. 9 shows the mean of processing time of each node in case that there is one person in the experimental space and the latency of the system is 200ms. Table. 1 shows the amount of data sending from each node. Node-D receives the largest amount of data, 600KB/frame, of all nodes. This data size requires 4.8ms for receiving via Myrinet. And the actual throughput of the system is about 20fps and 13fps in case of one person and two persons respectively. This means that the actual throughput is lower than the theoretical one calculated by adding the longest processing time (node-D) and the longest data-sending time (to node-D) owing to the overhead of OS such as process switching. However enough performance is realized by using a PC-cluster.

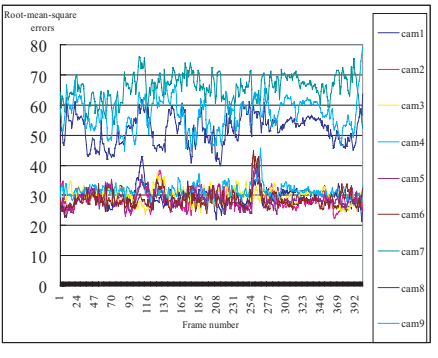


Fig. 8. Error: Cam 7 is ceiling camera unused for model coloring. Cam 8 and cam 9 are cameras unused for shape reconstruction and model coloring

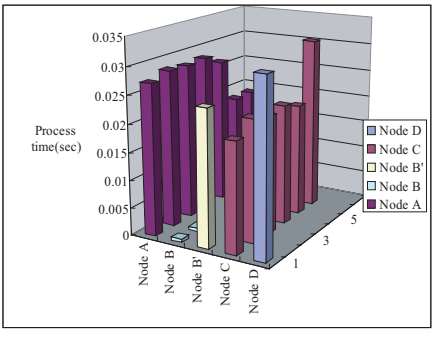


Fig. 9. Processing time from each node

4 Conclusion

In this paper, we propose a system generating free-viewpoint videos using multiple cameras and a PC-cluster in real-time. And we make some experiments to show the performance and the precision of our system.

Future works are as follows.

Reduction of latency. The latency of the current system is 200ms. That value is not small. We think that streaming processing and data compression is effective for reduc-

tion of latency. Streaming processing is introduced only from node-A to node-B' on the current system. So we will introduce streaming processing to all nodes.

Algorithm stable against object size. The throughput of our system is not stable since the number of voxels and the number of triangular patches depends on the size and the shape of objects. We think that variable resolution of the voxel space depending on the position of the virtual camera makes the throughput stable.

Precision of shape reconstruction. The precision of shape reconstruction much effects naturalness of free-viewpoint videos. Higher resolution of the voxel space, shape refinement after visual cone intersection and better object extraction from a camera image are planed to make.

References

1. P. J. Narayanan T. Kanade, P. W. Rander, "Concepts and early results," *IEEE Workshop on the Representation of Visual Scenes*, pp. 69–76, June. 1995.
2. Takashi Matsuyama, Xiaojun Wu, Takeshi Takai, and Shohei Nobuhara, "Real-time generation and high fidelity visualization of 3d video," in *Proc. of MIRAGE2003*, Mar. 2003, pp. 1–10.
3. Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel, "Free-viewpoint video of human actors," *ACM Trans. on Graphics*, vol. 22, no. 3, pp. 569–577, Jul. 2003.
4. Daisaku Arita and Rin-ichiro Taniguchi, "RPV-II: A stream-based real-time parallel vision system and its application to real-time volume reconstruction," in *Proc. of Second International Workshop on Computer Vision System*, Jul. 2001, pp. 174–189.
5. W. N. Martin and J. K. Aggarwal, "Volumetric description of objects from multiple views," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, no. 2, pp. 150–158, 1983.
6. Yukiko Kenmochi, Kazunori Kotani, and Atsushi Imiya, "Marching cubes method with connectivity," in *Proc. on International Conference on Image Processing*, Oct. 1999, vol. 4, pp. 361–365.
7. Roger Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Trans. on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.