Frame Rate Stabilization by Variable Resolution Shape Reconstruction for On-Line Free-Viewpoint Video Generation

Rui Nabeshima, Megumu Ueda, Daisaku Arita, and Rin-ichiro Taniguchi

Department of Intelligent Systems, Kyushu University,
6-1, Kasuga-koen, Kasuga, Fukuoka 816–8580, Japan
{nabeshima, ueda, arita, rin}@limu.is.kyushu-u.ac.jp

Abstract. Recently, the number of researches aiming at showing real world objects from arbitrary viewpoint have been steadily growing. The processing method is divided into three stages: 3D shape reconstruction by the visual cone intersection method, conversion of 3D shape representation from a voxel form into a triangular patch form, and coloring triangular patches. If the surface area of the object becomes larger, the frame rate decreases since the processing time of the conversion and coloring depends on the number of triangular patches. Stability of the frame rate is essential for on-line distribution of a free-viewpoint video. To solve this problem, we propose a new method which accommodates the space resolution during the 3D shape reconstruction step, thus stabilizing the number of triangular patches and the frame rate. This is achieved by raising the space resolution step by step and stopping the process on a time criteria. The reconstruction is done by using an octree-based visual cone intersection method. Experimental results show that this method makes the frame rate more stable.

1 Introduction

Currently, radio, television, etc. are used as telepresence. However, media using 3D information are more intuitive than those using 1D or 2D information like them because the world in which people live is 3D space. So, several researches have been done for generating the free-viewpoint video which shows objects in the real world from an arbitrary viewpoint using multiple cameras since Kanade et al. [1] had proposed the concept of "Virtualized Reality" [2][3]. However, they cannot generate the free-viewpoint video in real-time.

We are researching on on-line generation of free-viewpoint video[4], whose process consists of three stages:

1. reconstructing 3D shapes (voxel form) by the visual cone intersection method[5],
2. converting the voxel form of 3D shapes to the triangular patch form of them, and
3. coloring triangular patches.

As shown in the first stage, we reconstruct 3D shapes of objects explicitly. While, Matusik et al.[6] have proposed the image-based visual hull technique, which can generate a free-viewpoint video in real-time. This mathed generates a virtual viewpoint video by projecting rays from the viewpoint on image planes of cameras. This means that the

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 81–90, 2006.

methed does not reconstruct 3D shapes explicitly but implicitly and the amount of computation is smaller than explicitly reconstructing methods. However, since the method directly generates a virtual viewpoint video, the amount of computation is increased relatively to the number of virtual viewpoints. So, it is impossible for the method to deliver free-viewpoint videos to multiple viewers, each of whom can control his/her own virtual viewpoint. On the oter hand, our method can broadcast the triangular patch form of objects with color information, or a CG model, to all viewers and generate free-viewpoint videos on thire own terminals. This means that the amount of comuptation does not depend on the number of viewers.

However, there is a problem. When the surface area of objects becomes larger, the frame rate becomes lower since processing time of the second stage and the third one depends on the number of triangular patches. Stability of the frame rate is very important for on-line distribution of free-viewpoint videos since large jitters of on-line distribution requires longer queues for data transmission.

In this paper, we propose a new method to stabilize the number of triangular patches and, or the frame rate, by dynamically changing space resolution of 3D shape reconstruction. This is realized by using an octree-based visual cone intersection method[7] and raising its resolution step by step until the time allowed for one frame is over.

2 Free-Viewpoint Video

In this section, we show the configuration of the conventional free-viewpoint video system.

Generating a free-viewpoint video is quite time consuming. Therefore, we use a PC-cluster and RPV [8], which is a programming environment for a real-time image processing on a distributed parallel computer (such as a PC-cluster). Multiple cameras synchronized by an external trigger generator are placed in a convergent setup around the center of the scene. Each camera is connected to a PC.

First, object silhouettes are extracted from video frames captured by a camera via background subtraction and noise reduction. For each object silhouette, a visual cone, which is defined as the cone whose apex is the viewpoint and whose cross section coincides with the silhouette, is obtained. Visual cones are represented in terms of voxel space. Then, visual cones from multiple viewpoints are gathered and intersected to construct the shape of the object. Thereafter, 3D shape representation is converted from a voxel form to a triangular patch form by the discrete marching cubes method [9]. Then, visible vertexes of triangular patches are colored based on one camera image. Then, color information of all cameras are integrated. Finally, free-view point images are obtained with color information from a virtual viewpoint directed by the user.

3 Visual Cone Intersection Using Octree

Increasing the space resolution step by step by the visual cone intersection method which we used is difficult. Indeed this method must scan all voxels in the finest space resolution. Therefore, we use an octree-based visual cone intersection method proposed by Sato et al.

3.1 Octree

An *octree* is a tree to index three dimensions. Each node has either eight children or no children. This means that one voxel is divided into eight voxels when the space resolution is increased.

3.2 Pass Algorithm

Each voxel is classified into three types as follows.

White. The eight vertices are projected on the silhouettes
Black. The eight vertices are projected on the background
Gray. The other voxels (borders of silhouettes)

Voxels of a pre-defined size called initial voxels are classified into the above three voxel types. After that, only gray voxels are subdivided. Such classification and subdivision procedure is recursively executed until a pre-defined finest space resolution. This subdivision procedure from an also pre-defined coarse space resolution to the finest one is named *pass* and the algorithm is named *pass algorithm*.

3.3 Multi-pass Subdivision Algorithm

Since the intersection test described above is very simple, it sometimes ends in failure. Fig. 1 shows an example of that case. In the example, a voxel is projected on a narrow tip of a silhouette that avoids its vertices. This voxel must be classified as gray. Nevertheless, it is classified as black. This voxel is named *failed voxel*. Then, a Multi-Pass Subdivision Algorithm (see Fig. 2) is proposed to solve this problem. In Step1, the pass algorithm is applied to initial voxels. Afterward, failed voxels are detected in Step2. Failed voxels are subdivided and the pass algorithm is applied to the voxels obtained in Step3. Failed voxels are detected once again in Step2. As long as failed voxels are detected, Step2 and Step3 are repeated. If no failed voxels are detected, the process ends.

The way to detect failed voxels is as follows. We connect neighboring gray voxels in the finest space resolution, and thus we get the surface of a reconstructed geometry. If there is a white voxel neighboring a black voxel, the surface is not closed. Therefore, we regard the both voxels as failed voxels. This also means that there cannot be failed voxels in the finest space resolution. As a result, failed voxels can always be divided.

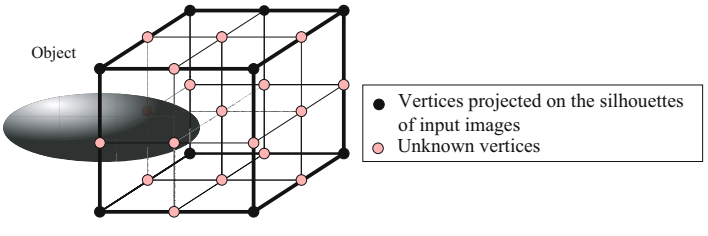


Fig. 1. Failed Voxel (Case Example)

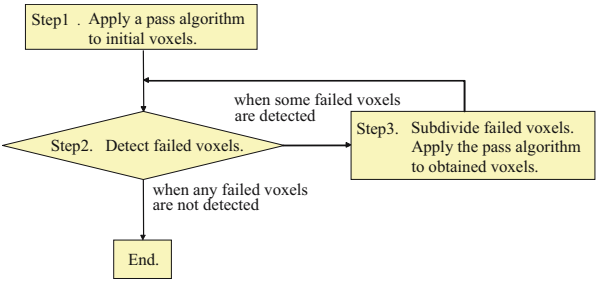


Fig. 2. Multi-pass Subdivision Algorithm

4 Frame Rate Stabilization by Variable Resolution Shape Reconstruction

4.1 Variable Resolution Shape Reconstruction

We use the octree-based visual cone intersection method to reconstruct the 3D shape in variable space resolution. Using this method, even if the process of 3D shape reconstruction does not reach leaf nodes of the octree, we can reconstruct the shape in lower space resolution using the information of the ancestor nodes. Then, we can stop the process and stabilize the frame rate.

The process of 3D shape reconstruction is shown in Fig. 3. The procedure from Step2 to Step4 is recursively executed until a certain time is over. When the time passes, the process stops at the end of either Step2 or Step3. We call the resolution in which the process is stopped *cutoff resolution*.

Step1. Apply the multi-pass subdivision algorithm in a certain cutoff resolution
Step2. Send leaf nodes in the current cutoff resolusion
Step3. Subdivide gray voxels and failed voxels in the higher cutoff resolution which cannot be subdivided in the previous subdivision
Step4. Apply the multi-pass subdivision algorithm

In Step1, when a node is judged to be black, the descendants of the node are considered as black nodes. Likewise, when a node is judged to be white, the descendants of the node are also judged to be white. Meanwhile, when a node, which cannot be divided in the cutoff resolution, is judged to be gray, the descendants of the node are judged to be gray.

In Step2, the information, white or black, of each leaf nodes is output. A gray voxel is output as a white voxel. However, when a gray voxel in the finest space resolution, the information of its vertices is output.

In Step1 or Step4, when there are failed voxels which cannot be divided in the current cutoff resolution but can be divided in the next higher cutoff resolution in Step3, they are retained and subdivided in the following Step3.

4.2 Detection of Failed Voxels

The first time we detect failed voxels, i.e. in the coarsest cutoff resolution, we scan all white voxels. For each white voxel, we detect black voxels which are minimum voxels

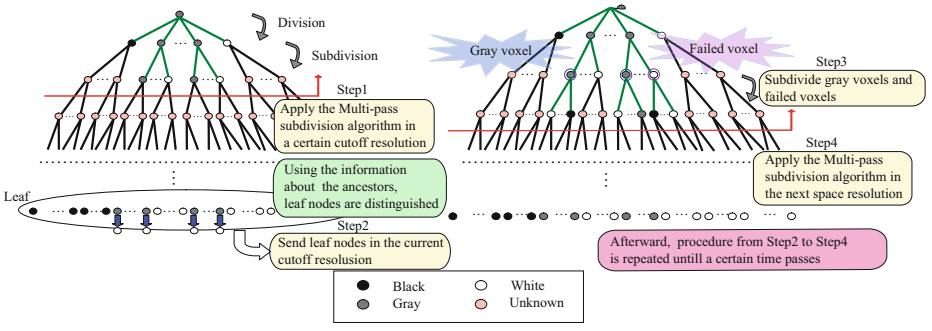


Fig. 3. Variable Resolution Shape Reconstruction

in the current cutoff resolution and are neighboring the white voxel. When we detect such a black voxel, we detect a black node which can be divided by tracing a link from the voxel to the parent node. If there is a node which can be divided, we regard the node as a failed voxel and divide it. In addition, if the white voxel can be divided, we also regard the node as failed voxel and divide it.

On the other hand, for the second time or later, we scan such white voxels and black voxels that are obtained by dividing gray voxels and failed voxels on the previous process, assuming failed voxels appeared at the time.

4.3 System Configuration

We obtained visual cones from each camera and intersected them in the conventional system. But we obtain visual cones from each three cameras and intersect them. By doing so, the surface area diminishes and the number of gray voxels decreases. It increases the number of voxels which do not have to be divided, and thus the processing time decreases.

The system configuration which we propose is as following. Processes are distributed to PCs shown in Fig. 4 and executed in pipeline parallel.

Node-A: Each node-A extracts object silhouettes from video frames captured by a camera by background subtraction and noise reduction, and sends the silhouette image to a node-B and a node-D. Each node-A sends a binary format silhouette image to a node-B to reduce the amount of data since each node-B uses the image to reconstruct a visual cone. On the other hand, each node-A sends a RGB silhouette image to a node-D since each node-D uses the image to color a visual hull.

Each node-A' does not works for model coloring but only for shape model reconstruction. Indeed model coloring needs large processing time, and it becomes difficult to use many cameras. The node-A's do not send images to node-Ds.

Node-B: Each node-B constructs a visual hull. When a-node-B refers vertices for voxel classification, it projects vertices onto the silhouette images from three viewpoints. Each vertex is regarded as occupied only when all projected point on each image are occupied. Each node-B sends the node-C a visual hull and information about the cutoff resolution one time or more.

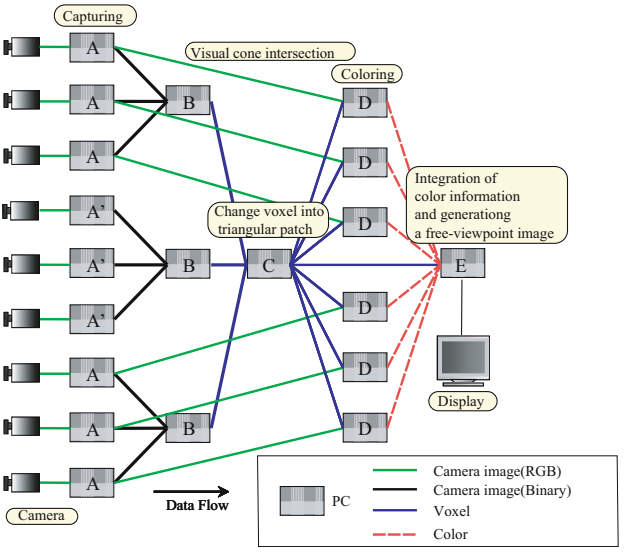


Fig. 4. System Configuration

Node-C: Each node-C intersects the visual hull every time it receives one from node-B. If the cutoff resolutions of all the visual hulls are not same, lower cutoff resolutions are adjusted to the highest cutoff resolution. Furthermore, the node-C converts the final shape model represented in terms of voxel space into triangular patches by the discrete marching cubes method. Then, the node-C sends the voxel space and its corresponding marching cube patterns to node-D and node-E since the data size of both voxel space and its marching cube patterns is smaller than that of all triangular patches.

Node-D and Node-E are the same as conventional system.

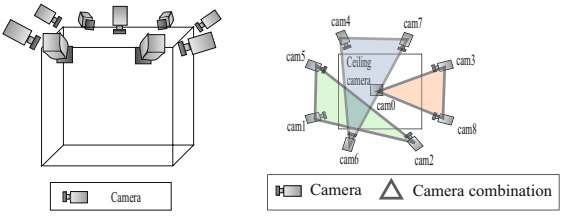
Node-D: First each node-D transforms the shape model represented in terms of a voxel space into triangular patches by using patterns sent from the node-C. Then, each node-D colors visible vertexes of the shape model based on one camera image. Finally, each node-D sends color information of all the vertexes of the shape model to the node-E.

Node-E: First, the node-E receives the position of the virtual viewpoint directed by the user. Then the shape model transforms into triangular patches in the same way as a node-D. Finally, the node-E integrates color information of all cameras and generates an image from the directed viewpoint.

5 Experiments

5.1 Experimental Environments

Using the system we proposed, we generate a free-viewpoint video in real-time to evaluate the processing time and the quality of generated images. We have used nine



(a) From a slanting top (b) From a top

Fig. 5. Camera arrangement and combination

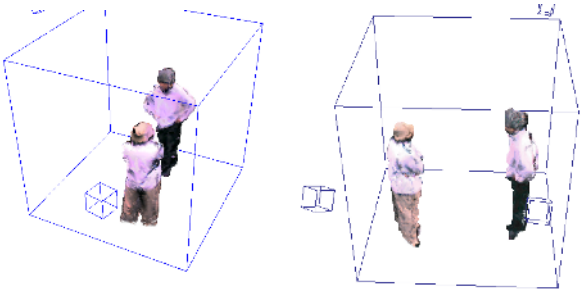
IEEE-based cameras at 320×240 pixel resolution, and 20 PCs (six node-As, three node-A's, three node-Bs, one node-C, six node-Ds, one node-E), each of which has an Intel Pentium4 (3GHz), 1GB memory and NVidia GeForce FX. PCs are connected with each other by Myrinet, a giga-bit network. All the cameras are calibrated in advance by Tsai's method [10]. The camera arrangement and combination of visual cone intersection in node-Bs are shown in Fig. 5. Maximal space resolution is $128 \times 128 \times 128$ and the size of a minimum voxel is $2cm$. The depth of the octree is five and the space resolution of initial voxels is $8 \times 8 \times 8$. The cutoff resolution is two level, $64 \times 64 \times 64$ and $128 \times 128 \times 128$. That is to say, we apply multi-pass algorithm until the depth of the octree is four and space resolution is $64 \times 64 \times 64$, and we advance to $128 \times 128 \times 128$ after sending leaf nodes.

5.2 Generated Free-Viewpoint Video

Fig. 6 shows camera images and generated images from a same viewpoint by the proposed variable resolution system and by conventional fixed resolution system. The space resolution of the fixed resolution system is $128 \times 128 \times 128$. In the variable resolution system, the shape reconstruction process is stopped at $64 \times 64 \times 64$ of the cutoff resolution since the total surface area of the objects, or two persons, is too large. Therefore, the object shapes look coarse because of the low space resolution. Moreover, the objects look bigger than those in the fixed resolution system, since we regard the gray

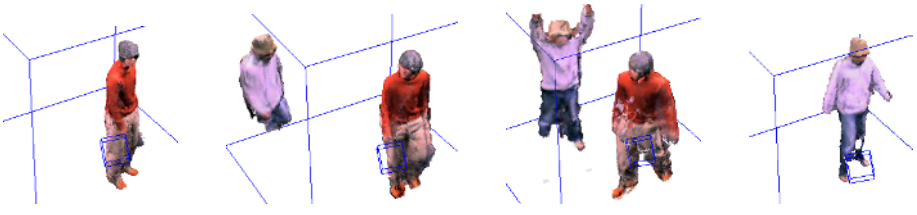
(a) Camera image (b) Generated image (Variable Resolution System) (c) Generated image (Fixed Resolution System)

Fig. 6. Camera image and generated image from a same viewpoint



(a) Generated image 1 (b) Generated image 2

Fig. 7. Generated images from virtual viewpoints



(a) Generated image 1 (b) Generated image 2 (c) Generated image 3 (d) Generated image 4

Fig. 8. Generated images in case that the number of persons varies

voxels in the reconstructed surface as white voxels. As a result, the surface of object is not colored correctly.

Fig. 7 shows generated images from virtual viewpoints by the variable resolution system. Small cubes in the images represent real camera positions. Small cubes represent camera position. The process is stopped at $64 \times 64 \times 64$ of the space resolution.

Fig. 8 shows generated image by the variable resolution system in case that the number of persons varies. When there are two persons, the process is stopped at $64 \times 64 \times 64$ of the space resolution. Otherwise, there is enough time to reach $128 \times 128 \times 128$ of the space resolution. This example shows the space resolution is changed according to the objects.

5.3 Processing Time

Not only the average of frame rate but also the variance are important for on-line free-viewpoint video distribution. Indeed, should the variance get higher, the communication buffer size will have increase as well, thus also increasing the delay time.

Fig. 9 shows frame rate in case the number of persons changes with the fixed resolution system (conventional technique) and with the variable resolution system (proposed technique). When the number of people changes from one to two, the frame rate of conventional system decreases significantly. On the other hand, the frame rate with pro-

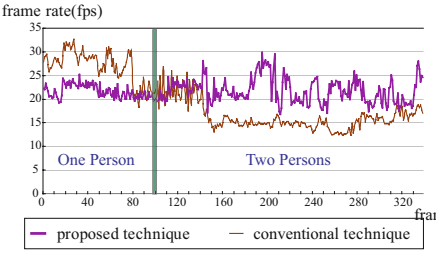


Fig. 9. Frame rate

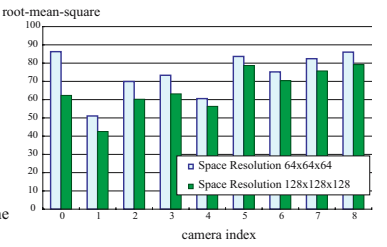


Fig. 10. Error of coloring

posed technique keeps 20fps independent of objects. Yet the frame rate with proposed technique has a small fluctuation. Indeed, stopping the process or not is judged after step2 or step3 (in section4) Stopping the process does not always happen fixed time.

5.4 Quality of Generated Image

When the process is stopped, space resolution is low, and the precision falls. We evaluate how different the error of coloring in space resolution $128 \times 128 \times 128$ and $64 \times 64 \times 64$ is. Fig. 10 shows the sum of root mean square error between a camera image and a generated image with the same viewpoint. The error in lower space resolution increase a little, but the increasing amount is much smaller than the error in space resolution $128 \times 128 \times 128$.

6 Conclusion

In this paper, we propose frame rate stabilization by variable resolution shape reconstruction for on-line free-viewpoint video generation. And we make some experiments to show the frame rate stabilization independent of the object. Major future works are as follows:

- reduction of the fluctuation in the frame rate
- compression of the amount of data transfer by sending not a voxel space but an octree
- realizing a multi-resolution marching cube method
- compression of color information
- developing an on-line free-viewpoint video distribution system.

References

1. T. Kanade, P. W. Rander, P. J. Narayanan:“ Concepts and early results”, IEEE Workshop on the Representation of Visual Scenes, pp.69–76, Jane 1995.
2. Shohei Nobuhara, Takashi Matsuyama:“ Heterogeneous Deformation Model for 3D Shape and Motion Recovery from Multi-Viewpoint Images”, Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, pp. 566–573, Thessaloniki Greece, September 2004.

3. J. Carranza, C. Theobalt, M. A. Magnor, H.-P. Seidel:“ Freeviewpoint video of human actors”, in ACM Trans. on Graphics, vol. 22, no. 3, pp.569–577, July 2003.
4. Megumu Ueda, Daisaku Arita, Rin-ichiro Taniguchi:“ Real-Time Free-Viewpoint Video Generation Using Multiple Cameras and a PC-Cluster”, Proc. of Pacific-Rim Conference on Multimedia, pp.418–425, December 2004.
5. W. N. Martin, 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. W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan:“ Image-Based Visual Hulls”, Proc. of SIGGRAPH, pp.369–374, 2000.
7. Hidenori Sato, Hiroto Matsuoka, Akira Onozawa, Hitoshi Kitazawa:“ Image-Based Photorealistic 3D Reconstruction Using Hexagonal Representation”, IPSJ Journal, Vol. 46, No. 2, pp.639–648, 2005.
8. Daisaku Arita, 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, pp.174–189, July 2001.
9. Yukiko Kenmochi, Kazunori Kotani, and Atsushi Imiya:“ Marching cubes method with connectivity”, in Proc. on International Conference on Image Processing, vol. 4, pp.361–365, Oct 1999.
10. R. Y. Tsai:“ A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses”, in IEEE Trans. on Robotics and Automation, vol. 3, no. 4, pp.323–344, 1987.