28th Picture Coding Symposium, PCS2010, December 8-10, 2010, Nagoya, Japan

PARALLEL PROCESSING METHOD FOR REALTIME FTV

Kazuma Suzuki†, *Norishige Fukushima*‡, *Tomohiro Yendo*†,
Mehrdad Panahpour Tehrani†, *Toshiaki Fujii*††, *Masayuki Tanimoto*†

†Graduate School of Engineering, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya-shi, Aichi, 464-8603 Japan
‡Graduate School of Engineering, Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya-shi, Aichi, 466-8555 Japan
††Graduate School of Engineering, Tokyo Institute of Technology,
2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550 Japan

ABSTRACT

In this paper, we propose a parallel processing method to generate free viewpoint image in realtime. It is impossible to arrange the cameras in a high density realistically though it is necessary to capture images of the scene from innumerable cameras to express the free viewpoint image. Therefore, it is necessary to interpolate the image of arbitrary viewpoint from limited captured images. However, this process has the relation of the trade-off between the image quality and the computing time. In proposed method, it aimed to generate the high-quality free viewpoint image in realtime by applying the parallel processing to time-consuming interpolation part.

Index Terms— FTV, Free Viewpoint Image Generation, Image Based Rendering, Realtime Processing, Parallel Processing

1. INTRODUCTION

We have proposed the Free Viewpoint Television (FTV) [1]. FTV is a television that enables us to watch from arbitrary viewpoint. If FTV is realized, we can freely move in the space inside the television. In this paper, we introduce a parallel processing method to generate free viewpoint image in realtime for FTV.

There are mainly two different approaches to generate arbitrary viewpoint images. One is the Model-Based Rendering (MBR) and this method consists of creating a 3D Computer Graphics (CG) model of the object inside the image. However, it is difficult to make a graphic model close to the real object. As a result, the created image appears to be artificial and unnatural. On the contrary, there is another method called Image-Based Rendering (IBR) [2, 3], which is not related to the geometrical characteristics of the object and can generate a photorealistic image from captured images. As one of IBR approach, there is the Ray-Space method [1], and this method records the position and direction of rays transmitted in the space. We have tried to apply the Ray-Space method to FTV.

The end goal of this research is to construct the FTV system in realtime. It is necessary to interpolate the image of a arbitrary viewpoint from the captured images to generate free viewpoint image for FTV. However, this process has the relation of the trade-off between the image quality and the computing time. Therefore, it very takes time to generate the high-quality free viewpoint image.

In this paper, we propose a parallel processing method to generate the high-quality free viewpoint image in realtime, and evaluate it. This paper is organized as follows. In the next section, we introduce a basic principle of free viewpoint image generation. In Section 3, we describe our proposed method to generate the high-quality free viewpoint image in realtime. The experimental results are explained in Section 4. Finally, in Section 5, conclusion and future work are described.

2. FREE VIEWPOINT IMAGE GENERATION

In this section, we introduce a basic principle of ray interpolation for free viewpoint image generation [4], and the method of depth estimation which is effective for the realtime processing.

2.1. RAY INTERPOLATION

For the free viewpoint image generation, an appropriate ray interpolation is needed. When we perform the interpolation, we assume that reflecting property of ray is Lambert reflection; the ray from the same light source must have same pixel value. If we can find the corresponded points of ray, we can interpolate the ray of the novel view using the pre-existing views.

This research is partially supported by Strategic Information and Communications RD Promotion Programme (SCOPE) 093106002 of the Ministry of Internal Affairs and Communications.

978-1-4244-7133-1/10/$26.00 ©2010 IEEE

Fig.1 shows this interpolation method. First, the viewpoint position(x, z) of a virtual camera is input, and the each pixel(u, v) is processed as follows.

1. The position of the interpolating ray on the reference plane are computed and the neighborhood 2 cameras are select -ed.
2. The depth of the ray source is changed, and the depth which can be regarded as Lambertian most is searched.
3. The intensity value is decided by performing linear interpolation according to the ratio of the distance of the ray on the reference plane.

The disparity of two cameras is assumed to be d, and output pixel value$VC(u, v)$ on the virtual camera is shown in the below.

$$VC(u, v) = \bar{a}LC(u + ad, v) + aRC(u - \bar{a}d, v)$$

This processing is conducted to all pixels(u, v), and the free viewpoint image is generated. In the expression above, $LC(i, j)$ and $RC(i, j)$ show pixel value of a left camera and a right camera of neighborhood.

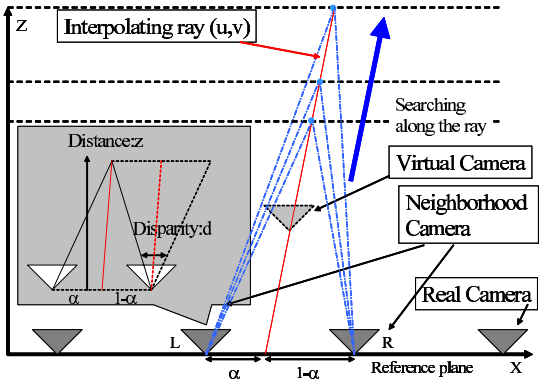

Fig. 1. Ray interpolation

2.2. DEPTH ESTIMATION

To estimate the object's depth, it is necessary to solve the correspondence point problem with acquired the neighborhood rays to interpolate the ray. If corresponded rays follow Lambert reflection, the neighborhood rays have almost same pixel intensity in a certain depth$d_{suitable}$ as shown below.

$$LC(d_{suitable}) \simeq RC(d_{suitable})$$

Thus, we can find corresponded rays using SAD (Sum of Absolute Differences) or variance. This assumption, however, is formed in only the ideal case because un-textured area do not have peak value, and in addition, CCD-noise make matching worse. Therefore we often use window based block matching to soften this effects. However, block matching have trade-off problem at the areas where un-textured and textured. In textured region, matching results have sharp peak. Thus, small block size of matching is better than large one for un-blur the result. On the contrary, in un-textured region, matching results have relaxed peak. Thus we must use large size block matching to reflect textured regions.

To solve this problem, the energy E of total balance between pixel consistency and depth smoothness is proposed because the object's depth changes smoothly. Minimizing this energy, un-textured region is optimized and more accurate depth can be obtained. The pixel consistency D is the matching result of pixel intensity, and the depth smoothness V is the penalty of depth discontinuity. λ is balance parameter.

$$E(d) = \sum_{p \in P} D_p(d_p) + \lambda \sum_{(p,q) \in N} V_{p,q}(d_p, d_q)$$

As one of the methods for solving this, the algorithm using Dynamic Programming (DP) is proposed [5]. DP algorithm has an advantage in computing cost over other proposed methods. Therefore, DP algorithm is used in this paper.

In DP, the depth smoothness is limited in the 1-D line. We assume that current pixel node depends on the pre-pixel node's depth d_{pre}. Thus we divide this problem into partial problem that minimizes the energy from starting pixel until x row's pixel. $F(x, d)$ is the energy of partial problem; d is the node value of depth. DP can be represented as recurrence equation. And we stack the computing result of previous best node value of depth $pd(x, d)$. Pixel consistency is measured by SAD and expressed $D(x, d)$.

$$F(x, d) = \min(F(x - 1, d_{pre}) + D(x, d) + \lambda |d_{pre} - d|)$$

$$pd(x, d) = \underset{d_{pre}}{\operatorname{argmin}}(F(x - 1, d_{pre}) + D(x, d) + \lambda |d_{pre} - d|)$$

Using DP method, the problem discussed above can be solved. However, even this method which computing cost is less than other methods have high computing cost for realtime processing.

3. PROPOSED METHOD

In this section, we introduce a parallel processing method to generate free viewpoint image in realtime. First, it explains the configuration and the flow of processing of proposed system. Then, the method of the parallel processing is discussed.

3.1. PROPOSED SYSTEM

The configuration of proposed system is shown in Fig.2 and the flowchart of proposed system's algorithm is shown in Fig.3.

In our system, 16 computers are used where each computer is responsible for controlling each camera. The 16 computers are called "client" as their operations are controlled by another computer called "server". The servers main task

includes user interface and client computer controlling. All the computers are connected forming a network under the Myrinet connection. The processing shown in Fig.3 is done by using this system.

First, the server sends the start signal to all clients. When the clients receive the start signal, all clients capture view images all together, and all captured images are sent to the server. After receiving the images, the server rearranges the image data to the data format that is suitable for the parallel processing for interpolation. This processing is described in the next paragraph. Then, the rearranged image data is sent back to each client, and the interpolation processing is conducted in parallel with each client. Finally, all interpolated image data is sent back to the server again and combined to generate the final new view image; it's free viewpoint image.

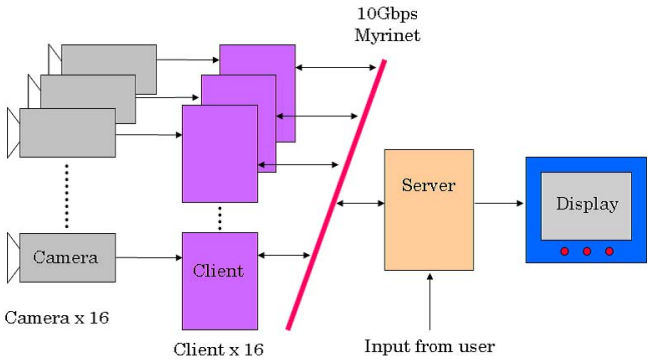

Fig. 2. The configuration of proposed system

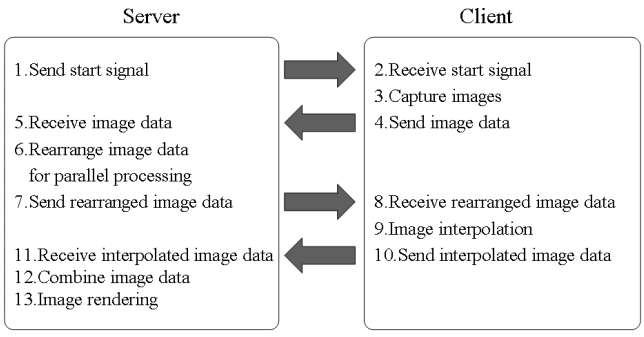

Fig. 3. The flowchart of proposed system's algorithm

3.2. PARALLEL PROCESSING

The interpolation method used here is very powerful but has some drawbacks on computational speed. Even for a small image size, the interpolation using one computer can hardly operate in realtime even at a low frame rate. To solve this problem, we propose the method of the parallel processing. In this process, the interpolation task is shared with all computers. In a word, each client computer individually interpolates a small section of the whole image data. The captured image data are then rearranged to the data format that is suitable for the parallel processing.

The rearrangement of data is based on two factors, the interpolation direction and the number of shared computers. The interpolation direction determines how image data are partitioned, while the number of shared computers determines the number of partitioning sections. For example, Fig.4 shows how data are rearranged when the interpolation is in horizontal direction using 4 shared computers. In this case, each input image is partitioned into 4 regions along the horizontal axis. The first regions of each image are grouped together and sent to computer 1 (PC1). The second regions of each image are grouped together and sent to computer 2 (PC2). In the same way, this process is repeated for the remaining regions and the remaining computers.

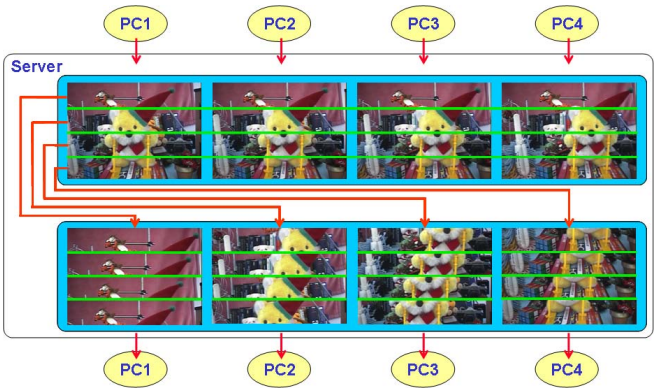

Fig. 4. Data rearrangement for parallel processing

4. EXPERIMENT

The experiment was conducted to evaluate the advantage of the proposed system. The high-quality free viewpoint image which PSNR exceed 37dB was generated and the computing time was measured.

4.1. EXPERIMENTAL CONDITION

Experimental conditions are as follows.

- 16 Camera images (Camera interval : 4cm, Image resolution : 640×480pixels, Camera array : 1-D array)
- 16 Computers (CPU : DualCore3.0GHz, Memory : 3.0GB)
- Network device : Myrinet (10Gbps)

4.2. EXPERIMENTAL RESULT

The generated image and the original image are shown in Fig.5. This image is not almost inferiority compared with the original image as shown in Fig.5.

Fig.6 and Table.1 show the comparison of the processing time between proposed method and conventional method (non-parallel). The image can be generated by using the proposed method at 66ms, though it takes 975ms with the conventional method. This corresponds to 15.2fps when it is made a frame rate, and the parallel efficiency is about 15 times.

Fig.7 shows the breakdown in each processing time when the free viewpoint image is generated by using the proposed method. For the whole computing time, the interpolation processing part accounts for 41%, the overhead (data communication) part by using the parallel processing accounts for 41% (receive image:17%, distribute image:18%, combine image:6%). The interpolation processing part and the data communication part account for about 80% of the whole computing time when both are added. It is necessary to improve these parts to make the frame rate higher.

Fig. 5. The generated free viewpoint image

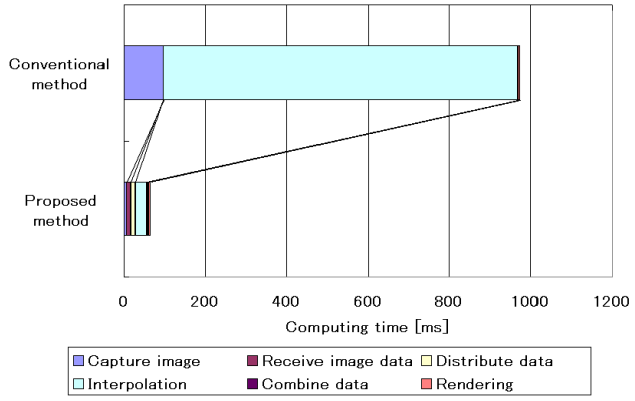

Fig. 6. The computing time of free viewpoint image generation

5. CONCLUSION

In this paper, we proposed the parallel processing method in realtime for free viewpoint image generation which has the relation of the trade-off between the image quality and the computing time. Using this method, the high-quality free viewpoint images can be generated at 15.2fps of frame rate. As a future work, we would like to reduce the computing time of the interpolation processing part and the data communication part, and generate the free viewpoint images at the video rate (30fps).

This work was partly supported by the Grand-In-Aid for YoungScientists (B) of Japan Society for the Promotion of Science underGrant 22700174.

Table 1. Each processing time of free viewpoint image generation

	Proposed[ms]	Conventional[ms]
Capture image	6	96
Receive image data	11	0
Distribute data	12	0
Interpolation	27	873
Combine data	4	0
Rendering	6	6
Total	66	975

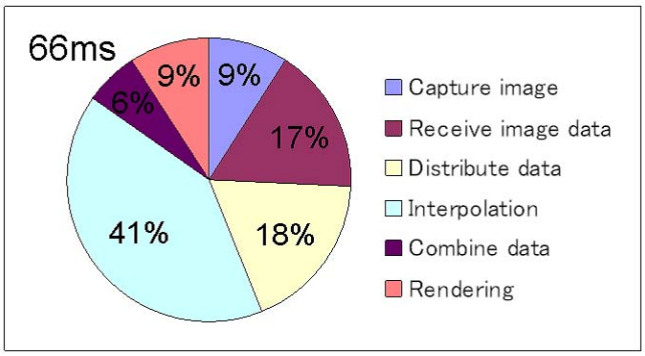

Fig. 7. The breakdown in each processing time of proposed method

6. REFERENCES

[1] T.Fujii, M.Tanimoto, "Free-viewpoint Television based on the Ray-Space representation", SPIE ITCom'02, pp.175-189, Aug.2002.

[2] M.Levoy, P.Hanrahan, "Light Field Rendering", ACM SIGGRAPH '96, pp.31-42, 1996.

[3] S.J.Gortler, R.Grzeszczuk, R.Szeliski,M.F.Cohen, "The Lumigraph", ACM SIGGRAPH '96, pp.43-54, 1996.

[4] N.Fukushima, T.Yendo, T.Fujii, M.Tanimoto, "An Optimization Method for Free View-point Image Generation using Ray-Space", Proc. of IWAIT2007, pp.896-901, 2007.

[5] Y.Ohta, T.Kanade, "Stereo by intra- and interscanline search using dynamic programing", IEEE Trans. PAMI, Vol.7, No.2, pp.139-154, 1985.