PAPER

Real-Time View-Interpolation System for Super Multi-View 3D Display

Tadahiko HAMAGUCHI[†*], *Nonmember*, **Toshiaki FUJII**[†,††], *Regular Member*, *and* **Toshio HONDA**[†,†††], *Nonmember*

SUMMARY A 3D display using super high-density multi-view images should enable reproduction of natural stereoscopic views. In the super multi-view display system, viewpoints are sampled at an interval narrower than the diameter of the pupil of a person's eye. With the parallax produced by a single eye, this system can pull out the accommodation of an eye to an object image. We are now working on a real-time view-interpolation system for the super multi-view 3D display. A multi-view camera using convergence capturing to prevent resolution degradation captures multi-view images of an object. Most of the data processing is used for view interpolation and rectification. View interpolation is done using a high-speed image-processing board with digital-signal-processor (DSP) chips or single instruction stream and multiple data streams (SIMD) parallel processor chips. Adaptive filtering of the epipolar plane images (EPIs) is used for the view-interpolation algorithm. The multi-view images are adaptively interpolated using the most suitable filters for the EPIs. Rectification, a preprocess, converts the multi-view images in convergence capturing into the ones in parallel capturing. The use of rectified multi-view images improves the processing speed by limiting the interpolation processing in EPI.

key words: *super multi-view, view interpolation, multi-view camera, adaptive filtering, rectification*

1. Introduction

Binocular stereoscopic displays using liquid-crystal shutter glasses and multi-view displays using lenticular films or parallax barriers are commonly used 3D displays [1]. These types of displays suffer two problems: a sense of incongruity between the focal position of the image and the perceived position of the image and tiredness after lengthy observation. A super multi-view display with a super high-density view overcomes both problems and is thus expected to find application as a stereoscopic display in which a natural stereoscopic view is possible. A super multi-view display using a focused light array has been developed [2].

The real-time view-interpolation system we have been developing captures stereoscopic images of objects, generates super multi-view images of them in real time, and sends the images to a super multi-view display. More specifically, we are developing a system that captures multi-view images of an object with a multi-view camera system, performs view interpolation and viewpoint conversion in real time on the basis of those images, and generates a super multi-view image.

The concept of the view-interpolation component was previously reported [3]. In this paper, we describe its implementation and evaluation. We describe the concept in the next section. Multi-view capturing with a multi-view camera system is described in Sect. 3. The view-interpolation algorithms, implementation, and evaluation are in Sect. 4. And a real-time view-interpolation system is in Sect. 5. We conclude with a brief summary in Sect. 6.

2. Concept

Our real-time view-interpolation system creates the total number of viewpoints for a super multi-view display after increasing views by roughly 5–10 times by capturing the object images with a multi-view camera system and generating images at viewpoints between the cameras by interpolation processing. This is the most promising method for generating super multi-view 3D images.

Most research on view interpolation has used a method whereby the three-dimensional shape, surface texture, and other physical data of the viewed object are faithfully reproduced, after which images are interpolated at intermediate viewpoints using CG techniques. Emphasis has been placed on accurately reproducing the data of the actual space rather than on high-speed processing and stability.

In the case of real-time view-interpolation processing for the super multi-view 3D display, interpolated images must be generated with high stability at all times, regardless of the shape or arrangement of the viewed object or the type of light source and so on. In addition, interpolation images must be generated at the video rate (30 frames per second). However, a real-time view-interpolation system has not yet been developed. That is the purpose of our research.

A characteristic pattern appears when multi-view images are layered on top of each other to form a 3D

Manuscript received March 12, 2002.

Manuscript revised June 26, 2002.

[†]The authors are with the Telecommunications Advancement Organization of Japan, Tokyo, 113–0001 Japan.

[††]The author is with Nagoya University, Nagoya-shi, 464–8603 Japan.

[†††]The author is with Chiba University, Chiba-shi, 263–8522 Japan.

[*]Presently, with Information Technology R&D Center, Mitsubishi Electric Corporation.

rectangular parallelepiped followed by cutting it into cross sections. Filter interpolation is the technique whereby image processing is performed on this pattern to obtain high resolution followed by generation of images of intermediate viewpoints. More specifically, this technique involves generating intermediate images by increasing the resolution of the ray-space data [4] in the direction of the parallax axis by using a filter. One approach to increasing the resolution is to generate interpolation images by enhancing the resolution of the epipolar plane images (EPIs) [5]. Processing that increases the resolution of these images in the longitudinal direction is equivalent to processing that generates intermediate images. Although filter interpolation is less susceptible to errors because there are no problems with error accommodation and lends itself to high-speed processing because it does not require search processing, it generates less accurate interpolated images compared with those of block matching. Moreover, it is unable to perform processing corresponding to local characteristics since only a single interpolation function is used. Filter interpolation is promising in terms of stability and interpolation speed. Furthermore, it will enable processing that includes noise removal and pre-processing and post-processing.

Adaptive filter interpolation is also promising because it combines the characteristics of both block matching and filter interpolation. This technique, developed at Nagoya University, involves preparing multiple filter sets in advance and switching filters corresponding to the local characteristics of the EPIs [6]. We use it in our view-interpolation system to generate intermediate viewpoint images between cameras in real time.

3. Multi-View Capturing

3.1 Multi-View Camera

We previously used a four-camera system [3]. To increase the number of views and to shorten the camera interval, we now use an eight-camera system (Fig. 1). The cameras have a 1/2-inch CCD and a 12-mm fixed-focal-length lens; they are installed on an optical rail. Their positions and poses can be adjusted manually, and the minimum camera interval is 32 mm.

When the optical axes of the cameras are parallel, that is, in parallel capturing, the epipolar line is parallel to the x axis (horizontal axis). EPI is obtained as a horizontal cross section of a pile of multi-view images. Our view-interpolation technique uses the character of this EPI in which an object's corresponding points are located on a straight line. Therefore, the processing area of view interpolation can be limited on EPI, and a total processing time can be cut down.

In parallel capturing, the base line of multi-view camera is made long to expand the viewing area. For an object image to enter the viewing field of each camera, the lens needs a large angle. Therefore, the resolution of the image is poor. To improve the resolution, it is desirable to captured in convergence so that the object image may always be positioned near the center of a viewing field. Therefore, the convergence-capturing multi-view images need to be transformed into parallel capturing images, like those taken by a parallel capturing system. This is called rectification [3].

3.2 Rectification and Rectification Matrix Correction

The procedures for rectification and rectification matrix correction are shown in Fig. 2. First, a calibration target with known feature-point coordinates is prepared. Each camera captures the target, and the internal parameter of each camera is calculated. Next, the cameras simultaneously capture the calibration target, and the external parameter of each camera is calculated using the image captured by that camera and its internal parameter. This calculation of the internal and external parameters is done using a method that compensates for lens distortion [7].

Next, the rectification matrix of each camera is calculated from these parameters, and rectification matrix correction, including global parallax correction, is performed [3]. The matrix is used to rectify each camera's image. There are two general approaches to rectification. The matrix operations for rectification can be

Fig. 1 Multi-view camera system (eight cameras).

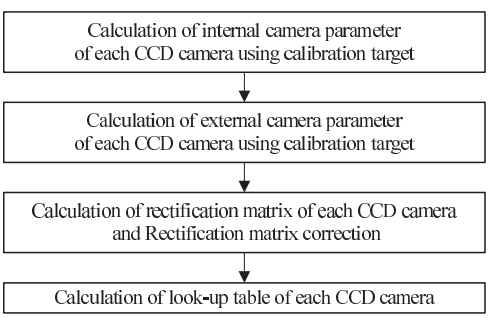


Fig. 2 Algorithm for look-up table calculation.

performed one by one, or a look-up table (LUT) can be calculated beforehand based on the rectification matrix and rectification is performed using LUT. We used the latter approach to enable real-time processing.

3.3 Real-Time Rectifier Circuit

A block diagram of the real-time rectifier circuit is shown in Fig. 3. The input and output images are in NTSC composite format and have 640×480 pixels. The LUT has 81×480 pixels , and interpolation is performed horizontally. Using the LUT, the images are converted at 60 fields per second. 17 LUTs can be stored in the rectifier circuit. Switching LUTs takes 0.3 seconds, so several fields of image disorder can occur at the time of switching. Since the purpose is to generate images for super multi-view displays, this disorder can be ignored. The LUT is transferred from an external host PC via an RS-232C interface. It takes about 30 seconds to transmit the data for one LUT.

4. Real-Time View Interpolation Algorithms

4.1 Filter-Selection Algorithms and Required Number of Filters

The algorithm for view interpolation using adaptive filters is illustrated in Fig. 4. Here, we describe the interpolation algorithm for color images. First, interpolation filters are prepared so as to be able to reflect the local characteristics of the EPIs. Next, each EPI is up-sampled in the longitudinal direction. The number of up-samplings is the same as the number of interpolation images needed. The R, G, and B color signals of the EPI are then transformed into Y, Cb, and Cr ones. Next, the pixels around the one to be interpolated are analyzed. This analysis is performed using the Y signal of the EPI. The optimum filter is selected based on the results of this analysis. Convolution processing is then performed on the Y, Cb, and Cr signals. Finally, these signals are transformed back into R, G, and B signals. This processing is performed on all pixels to be interpolated to obtain an output image.

Filter selection means choosing the filter that responded to the parallax between two original camera images in each pixel to be interpolated. The filters used are linear-interpolation filters. The two filter-selection methods used are shown in Fig. 5. In the variance minimization (VM) method, the convolution between each original camera image is calculated, as shown in Fig. 5 (a), and the filter that has the minimum variance between these convoluted images is chosen. When this variance is the minimum, the correlation between the EPI and the filter is the maximum. The block matching (BM) method searches for the parallax between two original camera images by block matching, as shown in a Fig. 5 (b). The pixel width used for block matching can be adjusted.

For view-interpolation processing using adaptive filters, we need to prepare a filter set in which the number of filters corresponds to the number of parallaxes between cameras. These parallaxes correspond to the depth of the object. The filter is required for the infinite number in order to correspond to depth until infinite from zero. However, this is not realistic. We thus considered only objects located in a limited depth, which limited the number of filters needed in the set. The relation between the depth of the object and the number of filters is given in Eq. (1), where f is the focal length, D is the camera interval, L is the distance between the multi-view camera and the object, b is the depth of the object, and p is the horizontal pitch of the CCD of the camera. The parallax between contiguous cameras is

$$d = \frac{fD}{pL} \tag{1}$$

When $L \gg b$, the range of the parallax, Δd, becomes

$$\Delta d = \frac{fD}{p}\frac{b}{L^2} \tag{2}$$

That is, the number of filters needed is only Δd. For example, for f=12 mm, D=30 mm, L=0.6 m, and

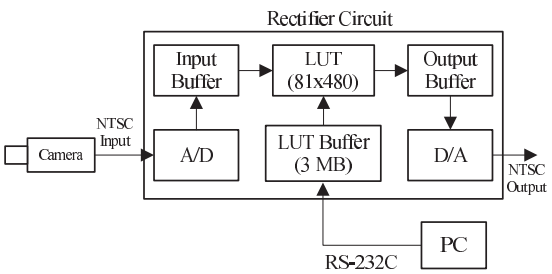


Fig. 3 Real-time rectifier circuit.

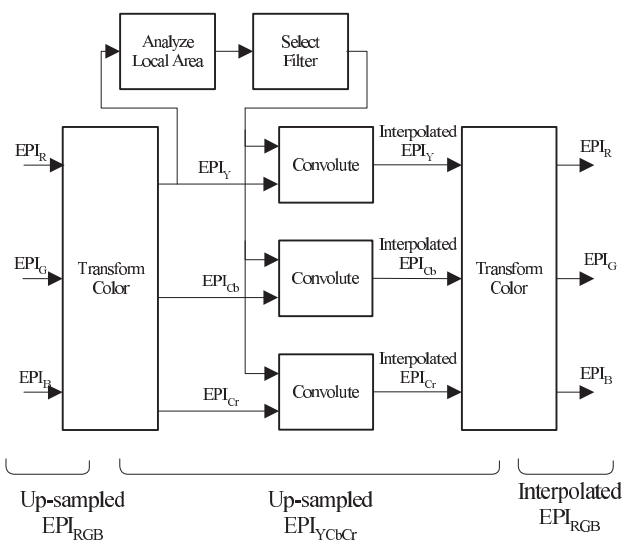
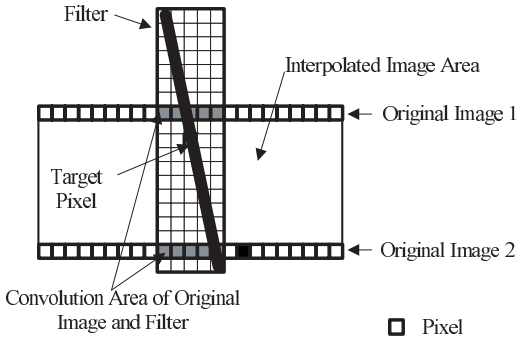
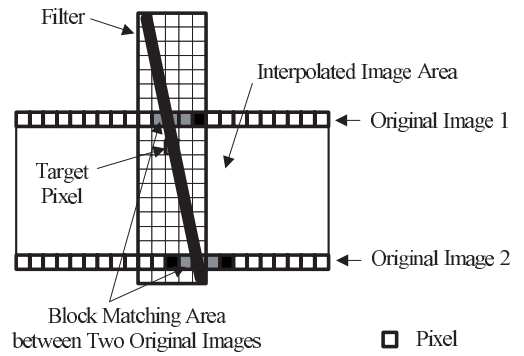


Fig. 4 Algorithm for view interpolation using adaptive filters.



(a) Variance minimization (VM method).



(b) Block matching (BM method).

Fig. 5 Two filter-selection methods.

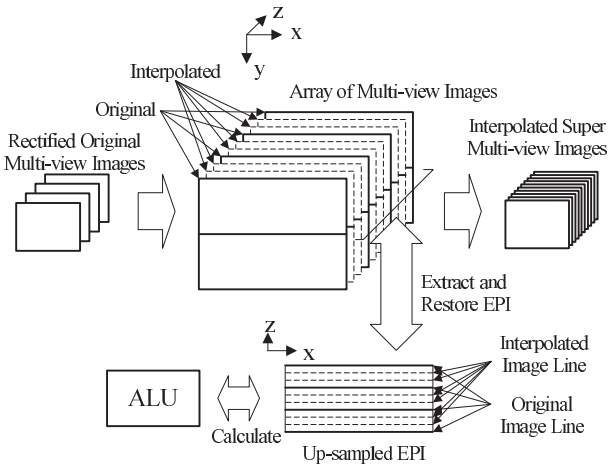


Fig. 6 Schematic diagram of view interpolation.

Table 1 Main specifications of parallel processor boards.

Board	Type 1	Type 2
Function	SIMD linear array processor	SIMD linear array processor
Multiplier	Included	Not Included
Number of PEs	32×8 chips	32×8 chips
Clock frequency	40 MHz	40 MHz
Internal memory	512 kB	256 kB
External memory	16 MB	16 MB

PE: processor element

p=0.022 mm (the case of a 1/2-inch CCD and 256-pixel horizontal resolution), if the depth is 50 mm, $\Delta d \sim 2.3$, so three filters are needed. We previously showed that the relation between the depth of an object and the number of filters needed is a parameter of the number of interpolations [3].

4.2 Implementation of View-Interpolation Processing

As mentioned above, we used the adaptive-filter interpolation method on EPI. From the external frame memory in which the multi-view image array is stored, one-line image data on a certain y coordinate of the rectified original image is transferred to internal memory in the processor, forming an EPI. Up-sampling of this EPI is performed according to the magnification multiple, and the arithmetic logic unit (ALU) in the processor interpolates the view and returns the result to an array of multi-view images. Once this processing is performed for all y coordinates, view-interpolation processing is completed. A block diagram of this processing is shown in Fig. 6. To optimize the algorithm in parallel with examination of a system configuration, we used programmable equipment for the processing system.

We previously implemented this system using a digital-signal-processor (DSP) board [2]. However, a single instruction stream and multiple data streams (SIMD) parallel processor board is a better development environment, so we used one in this work. The main specifications of this board are shown in Table 1. In the type 1 boards, the processor chips have a multiplier. In the type 2 boards, they do not. To determine which type of board is better, we tested them using the two filter-selection methods described above.

4.3 Evaluation of Parallel Processor Boards

We evaluated the view-interpolation processing ability of the two types of parallel processor boards using adaptive filters. To perform this processing on an EPI, we had to accurately rectify the original camera images. We used standard test images from the Multiview Image Database [8] of University of Tsukuba. Still images of a "stuffed toy" (a Santa Claus) were used. Three viewpoint images were parallel captured with a camera interval of 20 mm, an image format of 640×480 RGB 24 bits, a lens focal length of 10 mm, and a distance of about 70 cm between the cameras and the object.

The images were down-sampled in half resolution and picked up horizontally by 256 pixels and used as the original images for interpolation processing. The texture of the background was disregarded, and, since our aim was view interpolation of only the "Santa Claus" image, each original image was trimmed to only the "Santa Claus" so that the global parallax could be made zero.

The three original images are shown in Fig. 7 (a).

(a) Original images in three viewpoints.

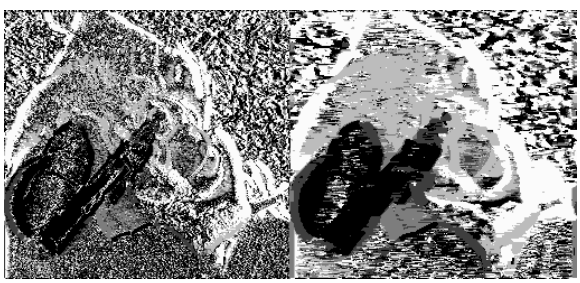
(b) Absolute differential image between Original 1 and Original 3.

Fig. 7 Original images.

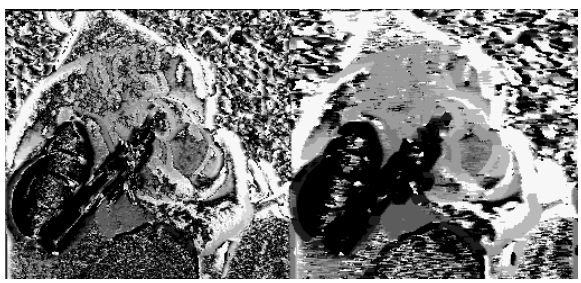
(a) Between 1 and 2.

(b) Between 1 and 3.

Fig. 8 Interpolated images (left maps are by VM method, right maps are by BM method).



(a) Interpolation between 1 and 2.



(b) Interpolation between 1 and 3.

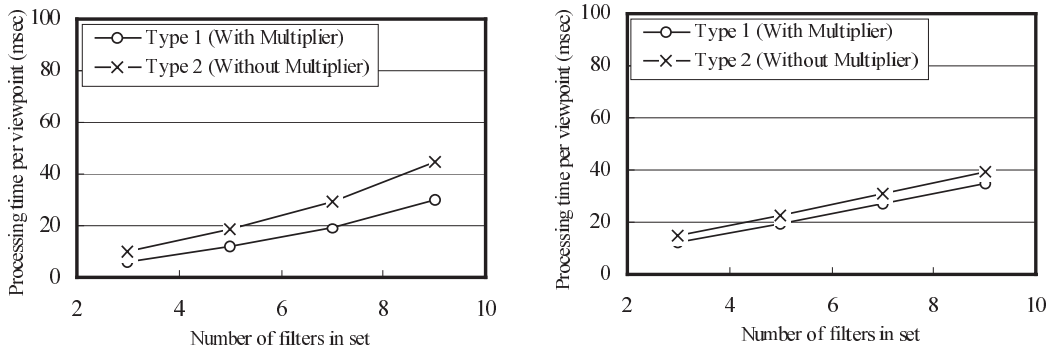
Fig. 9 Selected filter maps (left maps are by VM method, right maps are by BM method).

The absolute differential image between images 1 and 3, which reveals the parallax between them, is shown in Fig. 7 (b). As shown in this figure, the parallax was the highest around the image outline, where it was about six pixels. From this, the parallax between images 1 and 2 and between images 2 and 3 was about 3 pixels.

The images interpolated using the two filter-selection methods are shown in Fig. 8. Figure 8 (a) shows the images interpolated for the middle viewpoint between original images 1 and 2; the number of filters in the set was five. The left image shows the results of filter selection using the VM method, and the right image shows the results using the BM method. Figure 8 (b) shows the images interpolated for the middle viewpoint between original images 1 and 3; the number of filters in the set was nine.

Selected filter maps of the interpolations shown in Fig. 8 are shown in Fig. 9. These maps are equivalent to a depth map. The selected filter in the set is equivalent to the gray value in these images. The number of steps in the gray scale is equivalent to the number of filters (five steps in Fig. 9 (a), and nine in Fig. 9 (b)).

As shown in Fig. 9, the BM method had fewer selection errors. As shown in Figs. 8 (a) and 9 (a), the interpolated images had less noise when the interval between cameras was narrower (the maximum parallax was three pixels). The VM method had a larger filter selection error. However, in the low-frequency image areas, other than at the edges, a filter-selection error had little effect on the interpolated images. Therefore, the quality of the two interpolated images does not have a difference. On the other hand, as shown in Figs. 8 (b)



(a) VM method. (b) BM method.

Fig. 10 Measured processing time for view interpolation.

and 9 (b), when the camera interval was larger, a maximum parallax of six pixels, the image area with a large parallax produced noise. Such noise was particularly noticeable in the area of the candle held in the right hand and in the area of the champagne glass held in the left hand.

To evaluate quantitatively the quality of the interpolated images, we calculated the peak signal-to-noise ratio (PSNR). We used the middle image in Fig. 7 (a), "Original 2," as the original image and the two images in Fig. 8 (b) as the interpolated images. The method of calculation was as follows. A mask for extracting only the Santa Claus image from the original image was created manually. The mask was used on the interpolated images. The PSNR was calculated for only the extracted portion, which constituted about 73% of the entire image area. The PSNR of the BM method (31.8 dB) was better than that of the VM method (31.2 dB). Table 2 shows the calculated results of the PSNR of the two-filter selection method.

We measured the processing time for view interpolation. Figure 10 shows the processing time per viewpoint for the two filter-selection methods, with the number of filters in the set as a parameter. The processing time other than for interpolation processing, such as memory-to-memory transmission time, was disregarded by using block transmission inside the processor. The interpolation processing can be classified into two parts: filter selection processing and convolution processing using the optimum filter. The same convolution processing was used for both the VM and BM methods. With the VM method, the Type 2 board required about 1.5 times more processing time. Because, although convolution of a Y signal and all filters needs to be calculated when filter selection, since each processor of the Type 2 board does not have a multiplier, it is because processing has taken time. With the BM method, a multiplier is not needed when a filter is being selected for block matching between original images, so this method is better for the Type 2 board, which does not have chips with multipliers. Overall, there was not much difference in the evaluation results between Type 1 and Type 2 boards. With the Type 1 board, which has chips with multipliers, the processing time was approximately the same with either method. The parallel processing board described above was used for interpolation processing this time. But, when an optimum view-interpolation algorithm is defined in the future, a general-purpose image processor like a DSP board or a parallel processor board does not need to be used. That is, implementation of a lower-cost circuit specialized for interpolation processing using adaptive filters can thus be attained.

Table 2 PSNR of two-filter selection method.

Method	VM	BM
PSNR(dB)	31.2	31.8

5. Real-Time View-Interpolation System

A block diagram of real-time view-interpolation system is shown in Fig. 11. It is composed of multi-view cameras, real-time rectifiers, view-interpolator boards, etc. The eight cameras provide eight views, the camera interval is variable, and the minimum interval is 32 mm. Convergence capturing is supported. To show the observer a minimum number of super multi-view 3D images, two interpolation images per eye are generated, using four sets of view-interpolator boards. A parallel processor board is used for view interpolation. There are 256×240 pixels in each interpolated image. Since the interval between a person's eyes is about 60-70 mm, three CCD camera images are needed to generate the required parallax images, which means a minimum of three real-time rectifier circuits are needed. From the result of Fig. 10 in 4.3, since view interpolation is performed at the video rate (30 frames per second), seven or less filters are needed in the view-interpolator board. For the filter-selection algorithm, the BM method worked better for the evaluation images we used. Both algorithms, however, need to be evaluated using different test images in the future.

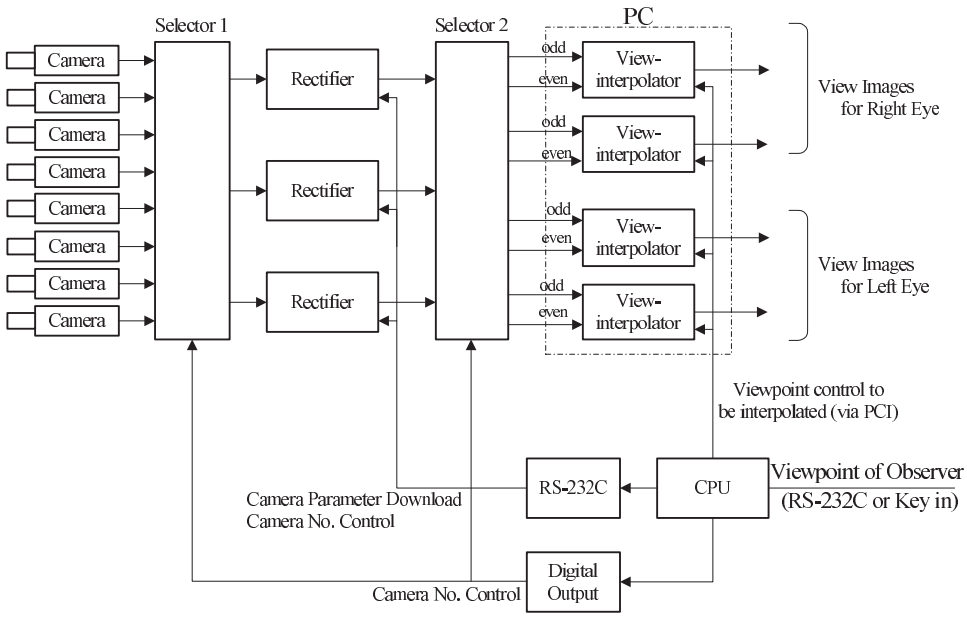


Fig. 11 Block diagram of view-interpolation system.

The flow of the processing is to first capture multi-view images using the eight cameras and then to select the three images best suited for the observer's viewpoint. The three selected images are rectified, and the view-interpolator boards generate two view images per eye (a total of four images). The two selectors used to select the three images are controlled by a digital output board in the host PC, and an appropriate LUT in each rectifier circuit is selected via an RS-232C connection.

6. Conclusion

A real-time view-interpolation system was developed for a super multi-view 3D display. View-interpolation processing was implemented in a parallel processor board, and this view-interpolation system can create an interpolated image in 256×240 format in real time. And a rectifier circuit that rectifies each NTSC format camera image in real time was implemented.

References

[1] T. Okoshi, Three-dimensional imaging techniques, Academic Press, 1976.
[2] Y. Kajiki, H. Yoshikawa, and T. Honda, "Three-dimensional display with focused light array," Practical Holography X, Proc. SPIE, vol.2652, pp.106–116, 1996.
[3] T. Hamaguchi, T. Fujii, Y. Kajiki, and T. Honda, "Real-time view interpolation system for super multi-view 3D display," Stereoscopic Displays and Applications VIII, Proc. SPIE, vol.4297, pp.212–221, 2001.
[4] T. Fujii, T. Kimoto, and M. Tanimoto, "Ray space coding for 3D visual communication," Picture Coding Symposium '96, vol.2, pp.447–451, 1996.
[5] R.C. Bolles, H.H. Baker, and D.H. Marimont, "Epipolar-plane image analysis: an approach to determining structure from motion," Int. J. Computer Vision, vol.1, pp.7–55, 1987.
[6] T. Kobayashi, T. Fujii, T. Kimoto, and M. Tanimoto, "Interpolation of ray-space data by adaptive filtering," Three-Dimensional Image Capture and Applications III, Proc. SPIE, vol.3958, pp.252–259, 2000.
[7] R.Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," IEEE J. Robotics and Automation, vol.RA-3, no.4, pp.323–344, 1987.
[8] Multiview Image Database, University of Tsukuba, 1995.

Tadahiko Hamaguchi received B.S. and M.S. degrees in Physics from Osaka University in 1982 and 1984, respectively. He joined Mitsubishi Electric Corporation in 1984. Since 1998, he has worked as a researcher on the Advanced 3-D Tele-Vision Project established by the Telecommunications Advancement Organization of Japan. His current research interests include 3D image capturing and processing. He is a member of the Institute of Image Information and Television Engineers of Japan.

Toshiaki Fujii received B.E., M.E., and Dr.E. degrees in Electrical Engineering from the University of Tokyo in 1990, 1992, and 1995, respectively. He is presently an Assistant Professor in the Graduate School of Engineering, Nagoya University. His current research interests include 3-D image processing and 3-D visual communications. He received an Academic Encouragement Award from the IEICE in 1996. He is a sub-leader of the Advanced 3D Tele-Vision Project established by the Telecommunications Advancement Organization of Japan. He is a member of the IEEE and the Institute of Image Information and Television Engineers of Japan.

Toshio Honda received a B.S. degree in Applied Physics from Waseda University in 1966, a M.S. degree in Control Engineering and a Dr.E. degree in Image Processing from the Tokyo Institute of Technology in 1968 and 1978, respectively. He worked as a research assistant and associate professor at the Tokyo Institute of Technology from 1968 to 1993. Since 1993, he has been a professor at Chiba University. His current research interests include optical methodology, 3-D image display, and image processing. He is a project leader of the Advanced 3-D Tele-Vision Project established by the Telecommunications Advancement Organization of Japan. He is a member of the Japan Society of Applied Physics, the Optical Society of Japan, and the Institute of Image Electronics Engineers of Japan.