# FREE VIEWPOINT IMAGE RECONSTRUCTION FROM 3-D MULTI-FOCUS IMAGING SEQUENCES AND ITS IMPLEMENTATION BY CELL-BASED COMPUTING


*Hiroki Yonezawa†‡, Kazuya Kodama†, and Takayuki Hamamoto‡*

†National Institute of Informatics
Research Organization of Information and Systems
Tokyo, Japan
E-mail: {clstyle, kazuya}@nii.ac.jp
‡Graduate School of Engineering
Tokyo University of Science
Tokyo, Japan
E-mail: hamamoto@isl.ee.kagu.tus.ac.jp



## ABSTRACT

This paper deals with the Cell-based distributed processing for generating free viewpoint images by merging multiple differently focused images. We previously proposed the method of generating free viewpoint images without any depth estimation. However, it is not so easy to realize real-time image reconstruction based on our previous method. In this paper, we discuss the method to reduce the processing time by dimension reduction for image filtering and Cell-based distributed processing. Especially, the method of high-speed image reconstruction by the Cell processor on SONY PLAYSTATION3(PS3) is described in detail. We show some experimental results by using real images and we discuss the possibility of real-time free viewpoint image reconstruction.



**Keywords:** image processing, image reconstruction, distributed processing


## 1. INTRODUCTION

In order to generate free viewpoint images from multiple differently focused images, conventional methods usually adopt depth-from-focus/defocus, and multi-view images are reconstructed from the depth information and the textures, that are obtained by merging detected focused regions in the original images. However, it is not so easy to estimate the depth information precisely enough to reconstruct free viewpoint images without artifacts. Especially, the depth of regions having no textures in the scene can be hardly estimated from multiple differently focused images, and such inaccurate estimation introduces awful artifacts in the reconstructed images [1].

Previously, we proposed the method of generating free viewpoint images directly from multi-focus imaging sequences without any depth estimation. The method realizes robust image reconstruction by applying a certain 3-D filter to the sequence. Then, we also proposed effective image reconstruction by integrating the 3-D sequence into a single 2-D image and applying a 2-D filter to the image [2]. It enables us to greatly reduce the cost of image filtering.

In this paper, we propose a method of high-speed image reconstruction by the Cell processor that is reasonable and gains high performance for media processing. We show some experimental results by using real images and we discuss the possibility of real-time free viewpoint image reconstruction.

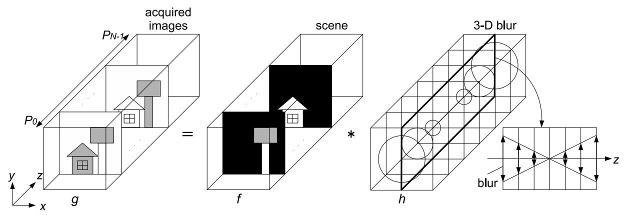



Fig. 1: A 3-D blur combines the scene and acquired images.

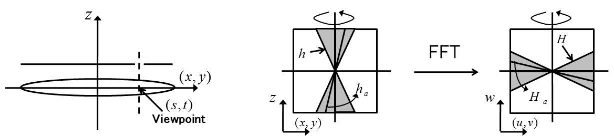



Fig. 2: A virtual lens and its 3-D filter.

## 2. FREE VIEWPOINT IMAGE RECONSTRUCTION

### 2.1 Free Viewpoint Image Reconstruction from 3-D Multi-Focus Imaging Sequences

As shown in Fig.1, multi-focus imaging sequence $g(x, y, z)$ is composed of acquired multiple differently focused images as 3-D information, and $f(x, y, z)$ corresponds to 3-D

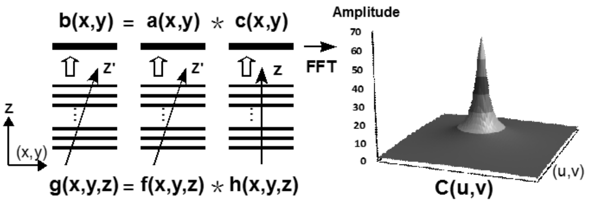
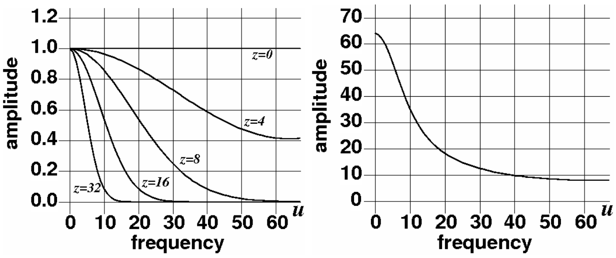



Fig. 3: Dimension reduction.




(a) $h(x, y, z)$ (b) $c(x, y)$

Fig. 4: Characteristics in the frequency domain.

information of a scene except occlusion. By using a convolution of a 3-D blurring filter $h(x, y, z)$, we can combine $f(x, y, z)$ and $g(x, y, z)$ in a space-invariant equation as Fig.1.

If blurs in acquired images are Gaussian blurs, the 3-D blurring filter in the spatial domain and in the frequency domain can be expressed as follows:

$$h(x, y, z) = p(x, y; r|z|/\sqrt{2}),$$
$$H(u, v, w) = Np(w; r|q|/\sqrt{2}),$$

where $r$ corresponds to the radius of the iris and $p(x; \sigma)$, $p(x, y; \sigma)$ corresponds to 1-D,2-D Gaussian function with the variance of $\sigma^2$ (we define $\lim_{\sigma \to 0} p = \ $ ), respectively. And, if the image size is denoted by $(N_x, N\ )$, $(K_x, K\ )$ $= (N/N_x, N/N\ )$ and $q^2 = K_x^2 u^2 + K^2 v^2$.

In case of $r = 0$, $h(x, y, z)$ corresponds to an ideal pin-hole camera, that is, a ray-set going through just the center of the lens.

Based on the characteristices of $H(r) = H(u, v, w; r)$, we previously proposed a method of generating an all-in-focus image $a(x, y, z)$ from the multi-focus imaging sequence $g(x, y, z)$ without any scene estimation as follows:

$$\begin{aligned} A(u, v, w) &= H(0.0)F(u, v, w) \\ &= H(0.0)H\ ^1(r)G(u, v, w). \end{aligned}$$

Here, we can define $H(0.0)H\ ^1(r)$ above as a single filter that uniquely exists and remains robust for all $(u, v, w)$. Therefore, generated images have good quality.

In this paper, we would like to also acquire images of the same scene by using a virtual pin-hole iris, whose position $(s, t)$ is different from the original one as shown in Fig.2. Its 3-D filter can be expressed as follows:

$$h_a(x, y, z; s, t) = \ (x + sz, y + tz),$$
$$H_a(u, v, w; s, t) = \ (w \quad (su + tv)).$$

Here, the virtual iris corresponds to a new viewpoint. The image sequence $a(x, y, z)$, that would be acquired by the iris can be expressed as follows:

$$\begin{aligned} A(u, v, w) &= H_a(u, v, w; s, t)F(u, v, w) \\ &= H_a(u, v, w; s, t)H\ ^1(r)G(u, v, w). \end{aligned}$$

If the virtual iris is in the area of the original lens, the filter $H_a(u, v, w; s, t)H\ ^1(r)$ remains also robust [3].

### 2.2 Efficient Free Viewpoint Image Reconstruction by Dimension Reduction

The cost of 3-D filtering described in the preceding section is not inexpensive, when the number of multiple differently focused images or the image size increases. In order to reduce the cost, here, we proposed the method realizing free viewpoint image reconstruction by a 2-D filtering [2]. First, we integrate acquired multiple differently focused images into 2-D information as shown in Fig.3. That is, we define new 2-D images and a new 2-D filter as follows:

$$a(x, y) = \textstyle\int f dz \ , \ b(x, y) = \int g dz \ , \ c(x, y) = \int h dz,$$

where $z$ -axis corresponds to a virtual viewpoint $(s, t)$. The axis may be different from $z$-axis. Then, we can obtain 2-D space-invariant equations as follows:

$$b(x, y; s, t) = c(x, y) \quad a(x, y; s, t),$$
$$B(u, v; s, t) = C(u, v)A(u, v; s, t).$$

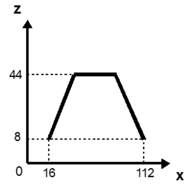
Here, we show the characteristics of $h(x, y, z)$ and $c(x, y)$ in the frequency domain for $r = 0.2$, $N_x = N\ = 128$, $N = 64$ in Fig.4, where $C\ ^1(u, v)$ uniquely exists and remains robust for all $(u, v)$. Therefore, we can reconstruct free viewpoint image $a(x, y; s, t)$ robustly as follows:

$$A(u, v; s, t) = C\ ^1(u, v)B(u, v; s, t).$$

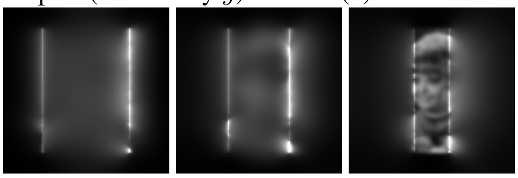
### 2.3 Simulations

We assume a scene that has a certain texture and various depths as shown in Fig.5(a), (b) , and a multi-focus imaging sequence (128×128 pixels, 64 images) of $r = 1.0$ is synthesized as shown in Fig.5(c). In case that a virtual viewpoint is inside of the original lens, based on the method described above, we realize image reconstruction robustly. Because of refraction, we need to correct the center of free viewpoint images. When the viewpoint is $(s, t)$, corrected images are obtained by translation of $\quad k(s, t)$ as shown in Fig.5(d), where $k$ is determined by camera parameters. The results indicate that we succeed in giving various disparities.

In order to clarify disparities, we generate a EPI(Epipolar Plane Image) of $y = 64$ from reconstructed free viewpoint images of various $s$ and $t = 0.0$ as shown in Fig.5(e). We notice that far objects correspond to slightly inclined lines, and near objects correspond to more inclined lines in the EPI. We can see appropriate disparities according to the depth information of the scene.

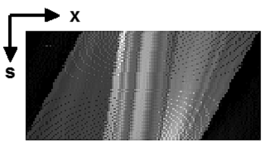


(a) Depths (at arbitrary $y$) (b) A texture



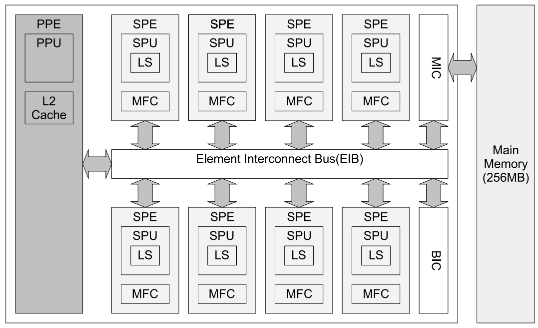
(c) Acquired images $g(x, y, z)$ $(z = 8, 16, 24)$

(d) Generated free viewpoint images

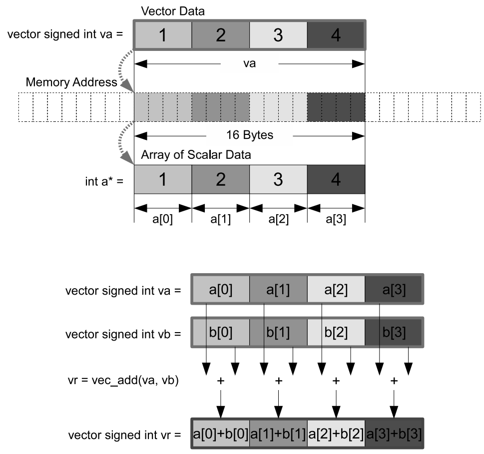



(e) EPI(Epipolar plane image) of $y = 64$

Fig. 5: Free viewpoint image reconstruction.




(a) Cell architecture




(b) SIMD(Single Instruction / Multiple Data)

Fig. 6: Cell architecture.

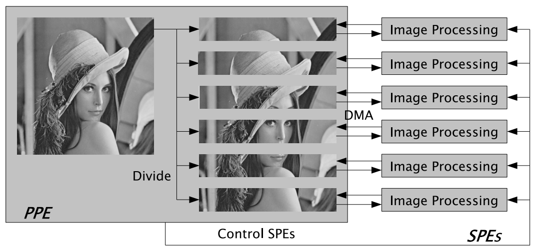



Fig. 7: Distributed image processing procedures on Cell.

## 3. CELL-BASED DISTRIBUTED PROCESSING

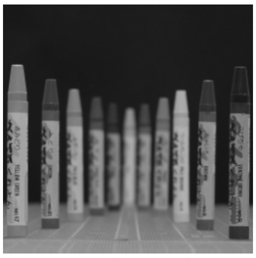
### 3.1 Reasonable and High-speed Processing Environment Using Cell Processors

As shown in Fig.6(a), Cell is the heterogeneous multi-core processor that is composed of 1 PPE(PowerPC Processor Element) and 8 SPE(Synergistic Processor Element)s [4]. PPE is the universal processor that has the same features as 64bits PowerPC architecture. Various operating systems run on the PPE and it controls I/O and the parallel arithmetic by SPEs. SPE is the 128bits high-performance processor. It is notable that SIMD (Single Instruction / Multiple Data) on PPE/SPE enables us to compute multiple 32bits floating point type data on single instruction as shown in Fig.6(b). In addition, DMA(Direct Memory Access) using MFC(Memory Flow Controller) allows high-speed data transfer between main memory and LS(Local Storage) that is the internal cache of SPE.

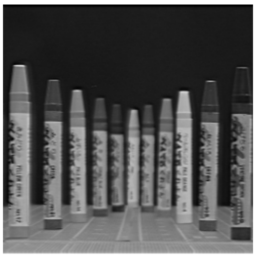
In this paper, we use Yellow Dog Linux 6.0 installed on PS3 as the platform for Cell-based computing. On PS3, one of 8 SPEs is made unusable by the factory default setting and the other one SPE is occupied by the Linux system. Consequently, we actually can use only 6 SPEs at the maximum for image processing.

### 3.2 Cell-based Distributed Image Processing

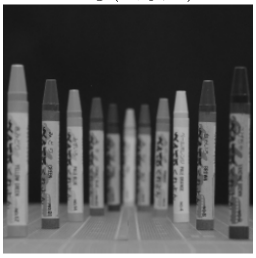
On the Cell processor, at first, PPE divides a target image into 6 sub-images for 6 SPEs as shown in Fig.7. Then, PPE transfers an appropriate binary program for required processing such as integration, transform and filtering to SPEs. Each SPE loads the divided image data from main memory into its LS with DMA. SPEs execute the image processing (integration and FFT/IFFT) and return result image data to main memory with DMA. Finally, each SPE notifies



(a) $g(x, y, 8)$

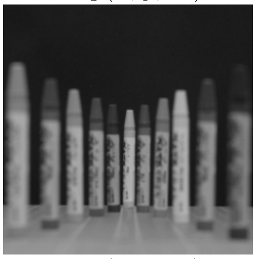


(d) $a(x, y)$, $left$

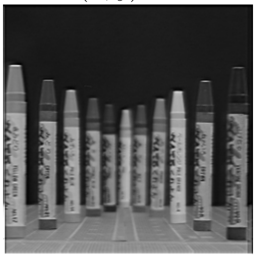


(b) $g(x, y, 16)$

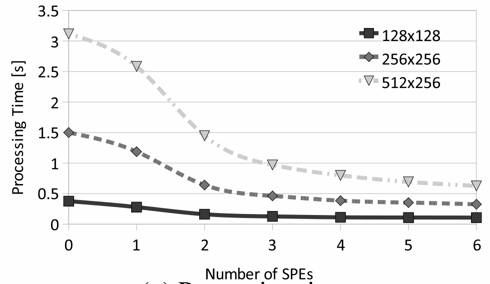
(e) $a(x, y)$, $center$



(c) $g(x, y, 24)$



(f) $a(x, y)$, $right$




(g) Processing time
(we use only PPE where the number of SPEs is 0)

Fig. 8: Free viewpoint image reconstruction from real images.

PPE of the completion of processing. Actually, SPEs process the image line by line when executing integration and FFT/IFFT, because of the restriction of LS memory size. After all, the procedure of our image reconstruction consists of (1) integration on SPEs, (2) FFT on SPEs, (3) filtering (multiplying) on the PPE and (4) IFFT on SPEs. Of course, we can obtain the same result as our previous method that does not use the Cell.

## 4. EXPERIMENTS USING REAL IMAGES

We experiment using a multi-focus imaging sequence of real images as shown in Fig.8(a)-(c). Each image has 256×256 pixels and the sequence consists of 32 images. In this experiments, the 3-D blurring parameter $r$ is estimated to be 0.36 by our previously proposed method [5].

In order to realize image reconstruction for the center viewpoint as shown in Fig.8(e), we spend 0.33 seconds. SIMD reduces the cost of integrating and filtering to about 1/4. For example, it reduces the cost of the filtering from 0.026 seconds to 0.008 seconds in case of 256×256 pixels. And distributed computing by using DMA reduces the cost of FFT and IFFT to about 1/4. Therefore, by using 6 SPEs, the whole of processing time can be reduced to less than 1/3. Next, we reconstructed free viewpoint images with various disparities by moving a viewpoint $(s, t)$ from left to right as shown in Fig.8(d)-(f). We can see that appropriate disparities are reconstructed.

As shown in Fig.8(g), we can realize image reconstruction, in case of 128×128 pixels, within about 0.1 seconds. Consequently, we expect about 10 fps image reconstruction.

## 5. CONCLUSION

In this paper, we discussed the Cell-based distributed processing for free viewpoint image reconstruction from multiple differently focused images. Our method is based on a simple scene-independent filtering, and so the hardware implementation as described in this paper is very effective for the acceleration. By the experiment, we show the possibility of real-time image reconstruction by using the Cell processor on PS3.

In the future, we would like to improve our method in order to realize real-time free viewpoint image reconstruction actually. For example, by using the imaginary part, the cost of FFT and IFFT can be reduced to 1/2. In addition, by adopting double-buffering loading/computing at the same time can be realized on SPEs. It enables us to expect more effective acceleration.

## 6. REFERENCES

[1] K. Kodama, et al. "All-infocus image generation by merging multiple differently focused images in three-dimensional frequency domain," PCM 2005, Lecture Notes in Computer Science 3767, pp. 303–314, 2005.

[2] X. Ou, et al. "Efficient free viewpoint image reconstruction from multi-focus imaging sequences based on dimension reduction," In Proc. of IEEE ISPACS 2007, pp. 152–155, 2007.

[3] K. Kodama, et al. "Free Viewpoint, Iris and Focus Image Generation by Using a Three-Dimensional Filtering based on Frequency Analysis of Blurs," In Proc. of IEEE ICASSP 2006, Vol. II, pp. 625–628, 2006.

[4] D. Pham, et al. "Overview of the Architecture, Circuit Design, and Physical Implementation of a First-Generation Cell Processor," IEEE Journal of Solid State Circuits, Vol. 41, No. 1, pp. 179–196, 2006.

[5] K. Kodama, et al. "Free Iris Scene Re-focusing based on a Three-Dimensional Filtering of Multiple Differently Focused Images," In Proc. of IEEE ICIP 2006, pp. 2033–2036, 2006.