Proc. Int. Conf. on Artificial Reality and Telexistence (ICAT2003), pp. 249 - 256.

ICAT 2003
December 3-5, Tokyo, Japan
THE VIRTUAL REALITY SOCIETY OF JAPAN

All in-Focus View Synthesis from Under-Sampled Light Fields

Keita Takahashi[†], Akira Kubota[‡] and Takeshi Naemura[†]

[†]The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo
113-8656, Japan
{*keita,naemura*}*@hc.t.u-tokyo.ac.jp*

[‡]Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA,
15213, USA
kubota@andrew.cmu.edu

Abstract

Light field rendering (LFR) is a fundamental method for generating new views from a set of pre-acquired images. We use densely-aligned cameras for the process of acquiring the set of images. In most practical cases, the density of the aligned cameras is not high enough to synthesize appropriate views. This "under-sampling" condition causes focus-like effects in the synthesized views. This paper proposes a new method for solving this problem. First, a set of differently *focused* views is synthesized from the under-sampled set of pre-acquired images. Then, an all *in-focus* view is generated from the set of differently *focused* views. This is based on a new *focus* measurement algorithm specialized for light field rendering and plenoptic sampling theory. Experimental results show the effectiveness of our approach.

Key words: Light Field Rendering, Focal Plane, All-in Focus, Focus Measure, Plenoptic Sampling

1. Introduction

In virtual reality and telecommunication systems, interactive viewing of photo-realistic 3D scenes is one of the most important technical issues. As a fundamental method of image-based rendering, light field rendering (LFR) [1] offers a promising solution for this issue. By this method, we can synthesize free-viewpoint images (synthetic views) from a set of pre-acquired images (input images) with little or no geometric model of the scene. It is possible to synthesize highly photo-realistic images in real time with graphics hardware accelerations.

The algorithm of LFR is simple. Each pixel value of input images is parameterized as a light-ray data, and stored in a light-ray database. We can synthesize a view of a given viewpoint from the database by picking up the date of such light-rays those pass through the viewpoint. In this process, light-ray data are interpolated on the assumption that objects in the scene are situated on a plane, which is called a *focal plane*.

In most practical cases, the set of input images cannot not be acquired densely enough to produce free-viewpoint images with adequate quality. In other words, light-ray data is under-sampled. This under-sampling condition causes focus-like effects[1] on the synthetic views [2]. Objects situated near the *focal plane* are synthesized clearly and sharply (*in focus*), while objects apart from the *focal plane* are with blurring and ghosting (*out of focus*). This is a serious problem in the sense of synthesis quality. Therefore, our concern in this paper is how to synthesize all *in-focus* images at arbitrary viewpoints without increasing the number of input images.

A main approach to that problem is utilization of geometric information of the scene in addition to the light-ray database [2, 3, 4, 5, 6]. In this approach, the structure of the scene is approximated by a *focal surface*, a set of depth layers or a 3D mesh model which represent the scene structure more presicely than a simple *focal plane*. In general, more precise geometry can generate more favorable synthetic views. However, these methods need a pre-estimation process to recover the scene structure before the image synthesis process. It consumes much time and computation power, and in some cases it needs some manual procedures. Even the most sophisticated computer vision technologies might fail to estimate the shape of objects with adequate quality, since their results are subject to the material of the objects and lighting conditions. Defectiveness of geometric models would cause highly visible holes and scums in the synthetic views.

In this paper, we propose a different approach from the ones mentioned above for synthesizing all *in-focus* images. First, for a given viewpoint, we synthesize some sequences of differently *focused* images by changing the depth of the *focal plane* of the LFR method. Then, we apply a new *focus* measure to detect *in-focus* regions in the differently *focused* images by comparing the sequences. Finally, we integrate them into an all *in-focus* image. This means that a view-dependent depth map is generated for each viewpoint. Thus, the proposed method does not use a global depth map that requires pre-estimation process. The merit of the view-dependent depth map is that it is good enough for the corresponding viewpoint.

Our approach has a close connection to the conventional depth from focus method [7, 8] or image fusion method [9], which utilize multiple differently focused im-

[1] In LFR, input images are assumed to have no defocus blurring. Notice that what is called *focus* here is an unique effect in LFR.

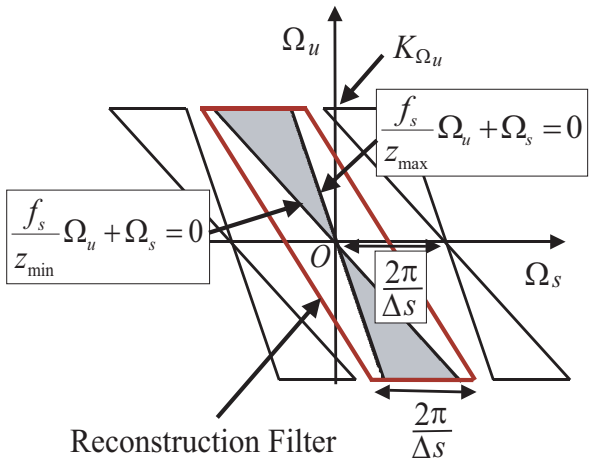


Fig. 1: Signal analysis of light fields in the frequency domain.

ages taken by physical cameras. However, the focus-like effects generated by the LFR method have different nature from the ones by physical cameras. For example, in the case of LFR, not only the blurring but also the ghosting effect appears in *defocused* regions. Therefore, we need a new *focus* measure that is specialized for LFR.

The remainder of this paper is as follows. In Section 2, related works are summarized to explain the phenomenon of *focus* in LFR from the theoretical viewpoints. In Section 3, we propose a new *focus* measure that is specialized for a set of images synthesized by the LFR method. We also describe a specific method for synthesizing an all *in-focus* image from the set of images using *focus* measurements. In Section 4, our implementations and experimental results are described. Finally we conclude our paper in Section 5.

2. Background

In this section, we introduce the theory of plenoptic sampling [10] that analyses light field signals in the frequency domain. This theory gives an insightful view to clarify the reason why the focus-like effect appears on the synthetic views by the LFR method [2, 11]. In this paper, we adopt the parameterization of light fields described in [2].

2.1 Spectrum of Light Field Signals

Input images are captured at 2D grid points on a plane which is called a *camera plane*. We consider that the optical axis of each camera is orthogonal to the *camera plane*. The position of a camera and a pixel on the camera are denoted by (s,t) and (u,v). Then, each light-ray captured by input cameras is uniquely determined by a set of 4 parameters as (s,t,u,v). For simplicity, we discuss the 2D subspace (s,u) of the original 4D light fields (s,t,u,v).

In the plenoptic sampling theory, light field signals are analyzed in the frequency domain. (Ω_s, Ω_u) denotes the frequency space corresponding to the (s,u) space. As described in [10], the signal spectrum appears only in the shadowed area in Figure 1. That is to say:

(1) When the range of depth (distance from the *camera plane*) of the scene is represented as $z_{\min} \leq Z \leq z_{\max}$, the spectrum is bounded by the following two lines:

$$\frac{f_s}{z_{\min}}\Omega_u + \Omega_s = 0, \quad \frac{f_s}{z_{\max}}\Omega_u + \Omega_s = 0, \tag{1}$$

where f_s denotes the focal length of the input cameras.

(2) the maximum frequency in the Ω_u axis is denoted as $K_{\Omega_u}(= 2\pi K_{f_u})$, which is determined by the complexity of the scene textures, the resolution of the input cameras or the resolution of the synthesized image.

Since the light field is sampled discretely, replications of the original spectrum appear in a constant interval as shown in Figure 1. When the distance between input cameras is Δs, the interval in the frequency domain is represented by $2\pi/\Delta s$. The original spectrum would not overlap the neighboring replications, when the following relation is satisfied:

$$\frac{1}{z_{\min}} - \frac{1}{z_{\max}} \leq \frac{1}{K_{f_u} f_s \Delta s}. \tag{2}$$

2.2 Focus in Synthetic Views

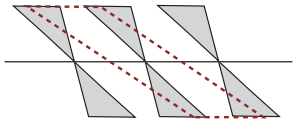
In the image synthesis process, the light-ray data are interpolated and re-sampled. To understand *focus* in synthesized images, we consider how to reconstruct a continuous signal from the discretely sampled signal.

In such cases that Equation (2) is satisfied, the ideal box filter shown in Figure 1 reconstructs the continuous signal without aliasing artifacts. The filter let through the entire spectrum components inside the box, while it completely cut off them outside the box. As described in [10], the slope of the box corresponds to the depth of the *focal plane*. Actually, the gradient of the sidewise lines of the box is $-z_0/f_s$, where z_0 denotes the depth of the *focal plane*. The filter is optimized when z_0 is equal to z_{opt} that is defined as

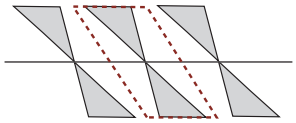
$$\frac{1}{z_{opt}} = \frac{1}{2}\left(\frac{1}{z_{\max}} + \frac{1}{z_{\min}}\right). \tag{3}$$

Shown in Figure 2 is the relationship between z_0 and the shape of the reconstruction filters. Even if Equation (2) is satisfied, when z_0 is much larger or smaller than z_{opt}, the continuous signal is not reconstructed properly as shown in Figure 2(a) and (c). The leakage of high frequency component and the interfusion of the neighboring replications cause blurring and ghosting artifacts in the synthetic views.

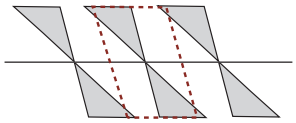
In most practical cases, Equation (2) is not satisfied for the whole scene since the sampling density $(1/\Delta s)$ is not high enough. In this paper, we consider such a small region of the scene where the depth variation is small enough to satisfy Equation (2). When the value of z_0 is optimized for the region, the light field signal of



(a) $z_0 < z_{opt}$



(b) $z_0 = z_{opt}$



(c) $z_0 > z_{opt}$

Fig. 2: Relation between the depth of the *focal plane* z_0 and the shape of the reconstruction filter.

the region is reconstructed clearly and sharply (*in focus*) without aliasing. In other words, when the value of z_0 is not optimized, the region is reconstructed *out of focus*.

3. All in-Focus View Synthesis from Under-Sampled Light Fields

In this section, we propose a new algorithm for synthesizing all *in-focus* views from under-sampled light fields. In our method, we first synthesize sequences of views by the conventional LFR method. What is important here is that their *focal planes* are different from each other's. Then, we integrate their *in-focus* regions to generate an all *in-focus* view. This is based on our proposal of *focus* measure specialized for LFR.

3.1 Focus Measure in LFR

In the case of physical cameras, defocus blurring can be modeled as the degradation of high-frequency components. Therefore, the high-frequency energy is used as a good measure of focus. Given a sequence of differently focused images, we could easily detect the in-focus region by searching such region that has the highest energy in the high-frequency domain among them. In the case of images synthesized by LFR, however, not only blurring but also ghosting artifacts that contain some high-frequency components arise in the *defocused* regions. Therefore, we need a novel *focus* measure specialized for LFR.

Our proposal for this *focus* measurement is to utilize the differences among images synthesized by different reconstruction filters which are to interpolate discretely-sampled light fields. Consider a set of images generated by different filters at an identical *focused* depth. The *focused* regions would be almost the same in all the images regardless which filter is used. The *defocused* regions, however, would show the differences of the characteristics of the filters. Therefore, subtractions of images generated by different filters can be used as the *focus* measure for LFR. That is to say, if a region is *in focus*, pixel values of the region are zero or very small in the subtraction image.

Now, we give a theoretical explanation on the principle mentioned above in frequency domain analysis. Consider such a small region that the depth variation is small enough to satisfy the following equation.

$$\frac{1}{z_{\min}} - \frac{1}{z_{\max}} \leq \frac{1}{2} \cdot \frac{1}{K_{f_u} f_s \Delta s} \tag{4}$$

The condition of this equation is more strict than that of Equation (2), since we need to use some other reconstruction filters in addition to the one described in [10].

In this paper, we use three different reconstruction filters shown as dashed parallelograms in Figures 3, 4 and 5. But, the number of filters is not limited to three in our method.

(1) normal filter (Figure 3): the most fundamental box-shaped filter. The width is $2\pi/\Delta s$. When z_0 is z_{opt}, it is the optimal filter described in [10].

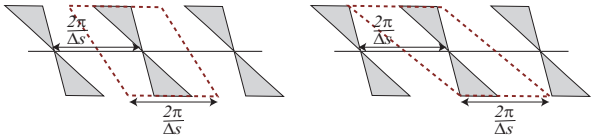
(2) wide-aperture filter (Figure 4) [2]: The width of this filter is smaller than $2\pi/\Delta s$. This leads to that the synthesized images would look like images taken by a wide-aperture camera. In this paper, the width is set to $\pi/\Delta s$ as shown in Figure 4.

(3) camera-skipped filter (Figure 5): The input images used for the reconstruction are skipped alternately. In this case, The repeating interval of the spectrum becomes $\pi/\Delta s$. We apply the normal filter, the width of width is $\pi/\Delta s$, for the reduced light-ray data.

When the region is *in focus* ($z_0 = z_{opt}$), all the filter would produce the same result, as shown in Figure 3(a), 4(a) and 5(a). When the region is *out of focus* ($z_0 \neq z_{opt}$), however, different filters would produce different results. This is because the frequency components passed through by the filters are obviously different from each other; the leakage of the original spectrum and the interfusion of the replication spectrum are caused differently as illustrated in Figure 3(b), 4(b) and 5(b).

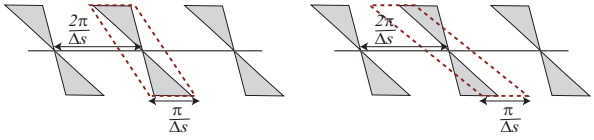
Now, consider a subtraction image between images synthesized by different kinds of filters for the same depth value of z_0. According to the Parseval's identity, the total energy of a signal is constant in the spatial domain and in the frequency domain. Therefore, the following conclusions are drawn from the frequency domain analysis.

- In a region that is *in focus*, every pixel in the region must be 0 theoretically.
- In a region that is *out of focus*, some pixels might have non-zero values.



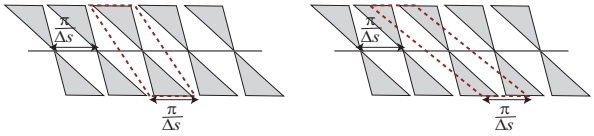
(a) $z_0 = z_{opt}$ (b) $z_0 < z_{opt}$

Fig. 3: Normal filter.



(a) $z_0 = z_{opt}$ (b) $z_0 < z_{opt}$

Fig. 4: Wide-aperture filter.



(a) $z_0 < z_{opt}$ (b) $z_0 < z_{opt}$

Fig. 5: Camera-skipped filter.

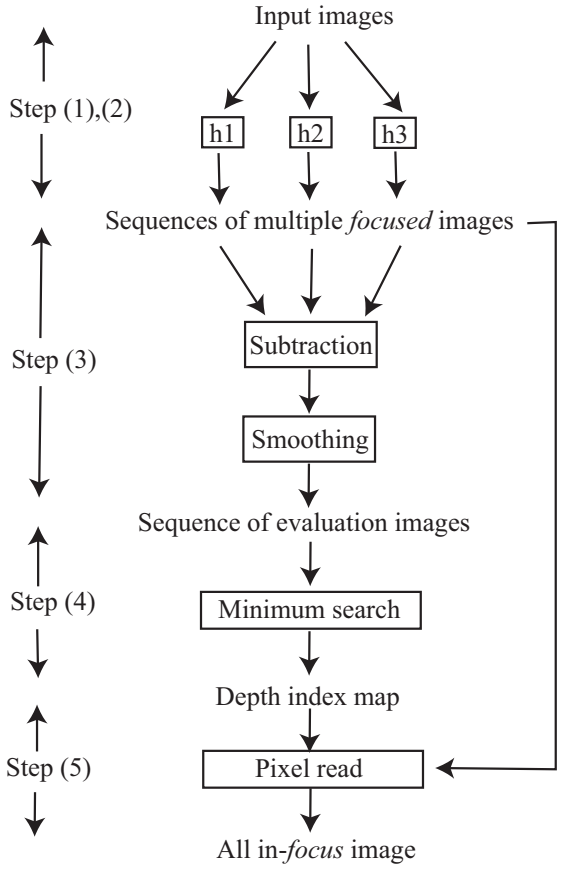


Fig. 6: Diagram of the proposed method.

3.2 All in-Foucs View Synthesis

First of all, we determine depth candidates z_n ($n = 0, 1 \ldots, N-1$) where the *focal plane* is placed, in such a way that each object would be *in focus* at least one image in the sequence. There are some useful theories [12, 13] for this configuration that discuss how much range of depth is *in focus* in front and rear of the *focal plane*.

The outline of our method is shown in Figure 6. The following steps are iterated for each novel view.

(1) The viewpoint for novel view synthesis is given.

(2) Three different sets of differently *focused* images, $I_k^{<n>}(x, y)$ ($n = 0, 1 \ldots, N-1$, $k = 1, 2, 3$), are generated by the LFR method corresponding to the filter h_k and the *focused* depth z_n. (x, y) denotes positions of pixels on the synthesized image.

(3) For each depth z_n, an evaluation image for the *focus* evaluation, $E^{<n>}(x, y)$, is calculated as follows:

$$E_0^{<n>}(x, y) = \sum_{k<l} a_{k,l} |I_k^{<n>}(x, y) - I_l^{<n>}(x, y)| \quad (5)$$

$$E^{<n>}(x, y) = \sum_{-M<i,j<M} E_0^{<n>}(x + i, y + j). \quad (6)$$

By taking combinations of more than two kinds of filters and summing $E_0^{<n>}(x, y)$ in a window region, the size of which is $(2M + 1) \times (2M + 1)$, we aim to enhance the stability of the *focus* measurement. $a_{k,l}$ denotes weighting coefficients.

(4) For each pixel (x, y), the index of the optimal depth where the pixel is mostly *in focus* can be given by $n(x, y)$ that yields the minimum of $E^{<n>}(x, y)$:

$$n(x, y) = \arg \min_{0 \le n \le N-1} E^{<n>}(x, y) \quad (7)$$

(5) An all *in-focus* image $I(x, y)$ is generated as follows.

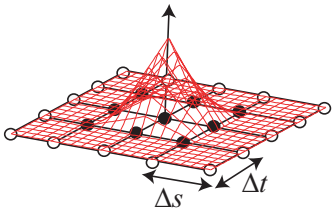
$$I(x, y) = I_k^{<n(x,y)>}(x, y) \quad (8)$$

In this way, each pixel value is read from the images synthesized in Step (2) where the pixel is mostly *in focus*, and an all in-focus images is generated.

4. Implementation and Experiments

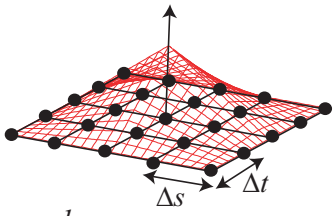
4.1 Implementation of LFR

First of all, this section explains our implementation of the LFR method, since our method consists of iterations of image synthesis by LFR. We adopt an implementation with texture mapping [4, 5], which can be accelerated by graphics hardwares. In this method, each input image is projected onto the *focal plane* as a texture data weighted



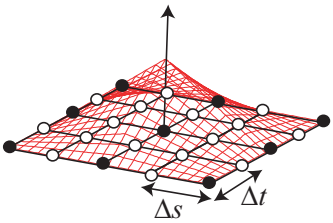
$$w(x,y) = (1-|x/\Delta s|)(1-|y/\Delta t|)$$

(a) Normal filter.



$$w(x,y) = \frac{1}{4}(1-|x/(2\Delta s)|)(1-|y/(2\Delta t)|)$$

(b) Wide-aperture filter.



$$w(x,y) = (1-|x/(2\Delta s)|)(1-|y/(2\Delta t)|)$$

(c) Camera-skipped filter.

Fig. 7: Weighting patterns for the filters.

Fig. 8: Results of free-viewpoint image synthesis by the LFR method. From the top to the bottom, the viewpoint moves from the right to the left while gazing at the nearest building. This leads to that the background moves from the right to the left.

by such a pattern that describes the contribution of each input image for the interpolation of light-ray data.

Shown in Figure 7 are weighting patterns adopted in our implementation and their formulas for each reconstruction filter. The weighting value is a function of position (x, y), having a peak at the center and decreasing to zero at the edges. Circles represent the grid points of the *camera plane* which denote the positions where input images were captured. For each texture data, the weighting pattern is aligned in such a way that its center is on the grid point where the corresponding input image was captured. Then, the weighting pattern and the texture data are multiplied and projected onto the *focal plane*. The above process is iterated for all the input images to synthesize an image.

Figures 7(a), (b) and (c) correspond to a normal filter, a wide-aperture filter and a camera-skipped filter, respectively. Note that these are approximations of the low-pass filters shown in Figures 3, 4 and 5. Those approximations are suitable for real-time implementations and sufficient to synthesize images with acceptable quality.

The above description corresponds to the step (1) and (2) in Subsection 3.2 which are executed on graphics hardware fast enough for interactive frame rates. Then, synthesized images are stored in main memory, and a CPU executes the remaining processes of Steps (3), (4) and (5).

4.2 Experiments

For our experiments, we use a Pentium 4 2.0 GHz, Linux based PC with 2.0 GB main memory. The video board loads a nVIDIA Geforce4 Ti 4200 GPU and 128 MB video memory. Our method is implemented with OpenGL enhanced by GLUT libraries.

A set of multi-viewpoint static images is used for input images, which is from "The multiview image database courtesy of University of Tukuba, Japan." These images are captured at 81 (9 by 9, in the horizontal/vertical direction) viewpoints. The interval of viewpoints is 8 mm.

(a) $z_0 = 300$ (around the nearest building).

(b) $z_0 = 600$ (around the furthest building).

(c) $z_0 = 1800$ (around the background wall).

Fig. 9: *Foucs* effects appear according to the depth of the *focal plane*. Regions near the *focal plane* are synthesized clearly and sharply, while regions apart from it seems blurry and noisy.

(a) Normal filter.

(b) Wide-aperture filter.

(c) Camera-skipped filter.

Fig. 10: Synthesized images by different reconstruction filters. *In-focus* region (further building) looks similar regardless of the filters, while *defocused* regions have different artifacts from each other.

The horizontal field of view is 27°, and the resolution is reduced from 640×480 pixels to 256×192 pixels for the convenience of our implementation. Distances from the *camera plane* to the nearest object, the farthest object and the background wall are 33 cm, 66 cm and 184 cm, respectively.

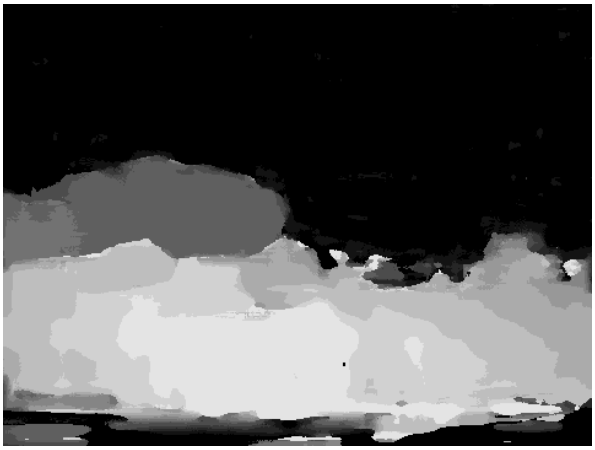
Shown in Figure 8 are synthesized images by the LFR method. Photo-realistic views are synthesized depending on the user's viewpoint interactively. But this conventional method suffers from the *focus*, which is noticeable in Figure 9. In Figure 9(a), (b) and (c), the *focal plane* is placed at 30 cm (around the nearest building), 60 cm (around the farthest building), and at 180 cm (around the background wall), respectively. Objects near the *focal plane* are synthesized clearly and sharply (*in focus*), while objects apart from the *focal plane* are with blurring and ghosting (*out of focus*).

To make everything *in focus*, the conventional method requires that the interval of input cameras should be smaller than 1.34 mm according to the plenoptic sampling theory [10]. It means that the number of input images needed for this scene amounts to 2916, which is 36 times (6 by 6) larger than that of the original input images. However, our method generates all *in-focus* images without requiring any increase in the number of input images.

All images mentioned above are produced by the normal filter. Shown in Figure 10 are synthesized images by the normal filter, the wide-aperture filter and the camera-skipped filter with the *focal plane* placed at 60 cm. They show the important fact that the *focused* regions are synthesized similarly regardless of the filters, while the *defocused* regions are synthesized differently according to the filter. This is the nature we utilize for *focus* measure as described in Section 3.

Now we describe the results of our proposed method. First, the number of depth candidates is set as $N = 13$, where in principle, all the objects in the scene would be *in focus* at least in one image of the sequences of differently *focused* images. The reconstruction filters h_1, h_2 and h_3 are assigned to the normal filter, the wide-aperture filter and the camera-skipped filter, respectively. We set that $a_{0,1} = a_{0,2} = 1$, and $a_{1,2} = 0$ in Equation (5), $M = 7$ in Equation (6), and $k = 1$ in Equation (8), are determined empirically. Their optimization is our future work.

Shown in Figure 11(a) is a depth map, each pixel of which represents the index of the optimal depth ($n(x, y)$ in Equation (7)). The brighter pixel indicates an object closer to the camera system. The structure of the scene is estimated approximately. This result is not so accurate for the use of the 3D reconstruction, but sufficient for the image synthesis. Figure 11(b) is a final image at the same viewpoint, it can be seen that all objects are reconstructed clearly and sharply in comparison with those in Figure 9. Note that we can synthesize images like this one at arbitrary viewpoints. It takes 1.5 – 2.0 seconds to synthesize an image with 512×512 pixels. There could be overhead costs in data transfer from video memory to main memory, and the processes on CPU are not well-optimized yet. Our future work includes optimization and implementation techniques to achieve interactive frame rates.



(a) Depth map.

(b) All *in-focus* image.

Fig. 11: The proposed method produces (a) depth maps and (b) all *in-foucs* images.

5. Conclusions

In this paper, we have proposed a free-viewpoint image synthesis method that overcomes the limitation of the conventional LFR method with no pre-estimated depth/ shape information. The proposed method generates all *in-focus* images from differently *focused* images synthesized by LFR. Our approach is reasonable since it utilizes the *focus* of LFR to estimate the depth, which is directly related to the appearance of synthesized images. We have also given a theoretical explanation on how to measure *focus* in the synthesized images by LFR. Experimental results show the effectiveness of the proposed method.

Acknowledgements

Thanks to Prof. Hiroshi Harashima of *the University of Tokyo* for his helpful discussion.

References

1. Marc Levoy and Pat Hanrahan. "Light Field Rendering", *Proc. ACM Conference on Computer Graphics (SIGGRAPH 96)*, pp. 31 – 42 (1996)
2. Aaron Isaksen, Leonard McMillan and Steven J. Gortler. "Dynamically Reparameterized Light Fields", *Proc. ACM Conference on Computer Graphics (SIGGRAPH 2000)*, pp. 297 – 306 (2000)
3. Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin and Werner Stuetzle. "Surface Light Fields for 3D Photography", *Proc. ACM Conference on Computer Graphics (SIGGRAPH 2000)*, pp. 287 – 296 (2000)
4. Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski and Michael F. Cohen. "The Lumigraph", *Proc. ACM Conference on Computer Graphics (SIGGRAPH 96)*, pp. 43 – 54 (1996)
5. Chris Buehler, Michael Bosse, Leonard McMillan, Steven J. Gortler and Michael F. Cohen. "Unstructured Lumigraph Rendering", *Proc. ACM Conference on Computer Graphics (SIGGRAPH 2001)*, pp. 433 – 442 (2001)
6. Takeshi Naemura, Junji Tago and Hiroshi Harashima. "Real-Time Video-Based Modeling and Rendering of 3D scenes", *IEEE Computer Graphics and Applications*, Vol. 22, No. 2, pp. 66 – 73 (2002)
7. Eric Krotkov. "Focusing", *International Journal of Computer Vision*, Vol. 1, No. 3, pp. 223 – 237 (1987)
8. Shree K. Nayer. "Shape from Focus System", *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 302 – 308 (1992)
9. Peter J. Burt and Raymond J. Kolczynski, "Enhanced Image Capture through Fusion", *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 173 – 182 (1993)
10. Jin-Xiang Chai, Shing-Chow Chan, Heung-Yeung Shum and Xin Tong. "Plenoptic Sampling", *Proc. ACM Conference on Computer Graphics (SIGGRAPH 2000)*, pp. 307 – 318 (2000)
11. Jason Stewart, Jingyi Yu, Steven J. Gortler and Leonard McMillan. "A New Reconstruction Filter for Undersampled Light Fields", *Proc. Eurographics Symposium on Rendering 2003*, pp. 150 – 156 (2003)
12. Yutaka Kunita, Masahiko Inami, Taro Maeda and Susumu Tachi. "Real-Time Rendering System of Moving Objects", *Proc. IEEE Workshop on Multi-View Modeling & Analysis of Visual Scenes (MVIEW'99)*, pp. 81 – 88 (1999)
13. Keita Takahashi, Takeshi Naemura and Hiroshi Harashima. "Depth of Field in Light Field Rendering", *Proc. IEEE International Conference on Image Processing (ICIP 2003)*, Vol. 1, pp. 409 – 412 (2003)