

Special Paper

방송공학회논문지 제24권 제7호, 2019년 12월 (JBE Vol. 24, No. 7, December 2019)

<https://doi.org/10.5909/JBE.2019.24.7.1246>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

Focal Stack Based Light Field Coding for Refocusing Applications

Vinh Van Duong^{a)}, Thuong Nguyen Canh^{b)}, Thuc Nguyen Huu^{a)}, and Byeungwoo Jeon^{a)‡}

Abstract

Since light field (LF) image has huge data volume, it requires high-performance compression technique for efficient transmission and storage of its data. Camera users may like to represent parts of image at different levels of focus at their choice anytime. To address this refocusing functionality, in this paper, we first render a focal stack consisting of multi-focus images, then compress it instead of original LF data. The proposed method has advantage of minimizing the amount of LF data to realize the targeted refocusing applications. Our experiment results show that the proposed method outperforms the state-of-the-art for LF image compression method.

Keywords: Light field compression, video compression, focal stack

1. Introduction

Light Field (LF) camera benefits from computational photography techniques which allow to capture not only the light intensities but also the light directions coming to the camera. It can provide many interesting functionalities like digital refocusing, depth estimation, viewpoint change,

and 3D reconstruction, etc. Ever since the first introduction of the LF concept by G. Lippmann in 1908 [1], it has kept being investigated. For example, H. Ives developed the first digital LF recording device in 1968 [2], M. Levoy applied the LF technology to microscopy in 2006 [3], and K. Fife designed an LF CCD sensor in 2008 [4]. In 2012, Ren Ng has built a hand-held LF camera called Lytro camera [5]. It has a micro-lens array placed before the sensor in order to make it act like a micro camera array that records multiple views of the same scene having slightly different perspectives each other. However, the multiple functionalities of LF require much data to be stored (or transmitted). For example, the raw lenslet image taken by Lytro Illum camera is of a size of 7728x5368, and its data in 4D LF format (i.e., multi-view images) contains 225 views (or sub-aperture images (SAIs)) of 625x434 pixels each. Thus, one single LF image is about 200 Mbytes which is more than

a) Department of Electrical and Computer Engineering, Sungkyunkwan University, Korea.

b) He performed this work while in Department of Electrical and Computer Engineering, Sungkyunkwan University, Korea. He is now with the Institute for Dataability Science, Osaka University, Japan.

‡ Corresponding Author : Byeungwoo Jeon

E-mail: hjeon@skku.edu

Tel: 82-31-290-7186

ORCID: <https://orcid.org/0000-0002-5650-2881>

※ This work was supported in part by the National Research Foundation of Korea (NRF) grant (2017R1A2B2006518) funded by the Ministry of Science and ICT.

· Manuscript received October 21, 2019; Revised December 13, 2019;

Accepted December 13, 2019.

30~40 times larger than a conventional image. Therefore, a powerful compression method is strongly required.

Many researchers have focused in past years on designing LF compression schemes. Some of recently noted are JPEG Pleno standardization on LF imaging in 2015 [6], ICME 2016 grand challenge on LF image compression [7], ICIP 2017 grand challenge on LF image coding [8], and ICME 2018 grand challenge on densely sample LF reconstruction [9]. Most of the methods reported so far can be basically divided into two approaches: a) to compress the lenslet image by exploiting the redundancy within its structure; and b) to compress LF data in the 4D LF format by further exploiting the redundancies in multi-view images. However, very few attentions have been paid to their compression schemes for specific application(s) of LF in mind. That is, most developments have focused only on how to maximally reduce size of the data in general. In [10], one study is carried out regarding the impact of light field compression on the focal stack and its extended focused images. The study showed that refocused image (RF) is more robust to blur than the extended focused image when it is affected by compression. Perra et al. [11] showed how sub-sampling of light field data before compression affects the end-users' experience of quality in light field applications. By down-sampling the light field image at an appropriate ratio, one can achieve better compression without much loss of

subjective quality. A comprehensive study about the perceptual quality of compressed light field and video streaming is also presented in [12] and [13], respectively.

In this paper, we aim to design a new compression scheme that specifically targets for refocusing application only. LF image allows one to be able to change the image focus after the image is already taken, but it requires a large amount of data for the process to be possible at user's side. Normally, we compress the 4D LF data first, and then render a set of its images focused at different depths (note that the set is called a focal stack). This approach can reduce the amount of data but not too much. By knowing a specific application, one can better render the focal stack images first and then compress its images instead of 4D LF data.

In literature, the focal stack is shown to be useful for estimating depth or creating an all-in-focus image (AIF). In [14], A. Levin et al. shows that one can fully reconstruct the 4D LF from a focal stack. This means that on one hand, it can reduce a great amount of data to be compressed by converting 4D LF data to a focal stack, but on the other hand, it can still recover the original 4D LF for different applications. Slightly different from this general goal, in this paper, we only target for the refocusing fidelity in LF applications. That is, we assume that users only want to see their object of interest at different depth levels. For this specific scenario in mind, we first render a focal stack

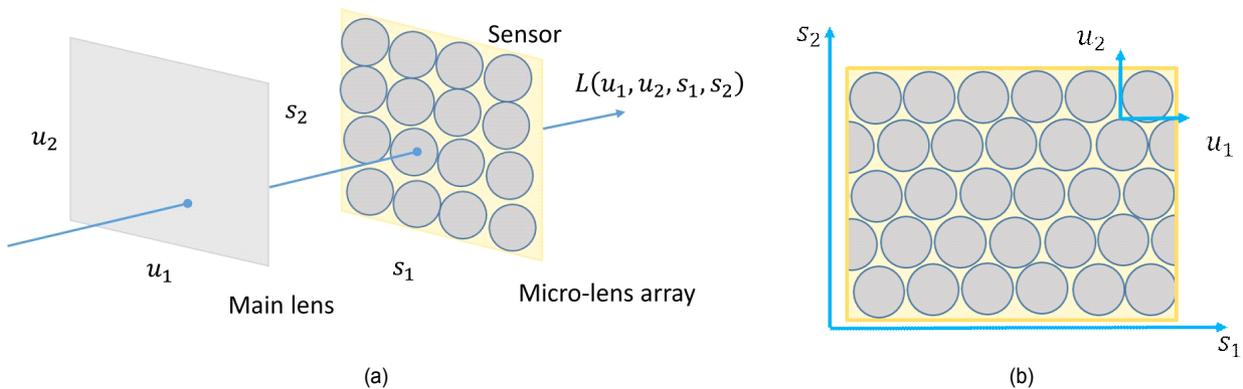


Fig. 1. (a) Light field representation as a 4D function $L(u_1, u_2, s_1, s_2)$; (b) All the light directions hitting the main lens at a coordinate (u_1, u_2) are recorded in the sensor behind each micro-lens. The spatial coordinate (s_1, s_2) corresponds to micro-lens array

which contains all the depth layers, then compress the focal stack directly. By doing so, it is expected that one can minimize the amount of data for transmission and storage. It should be noted that the number of refocused images (RFI's) is much smaller than that of SAIs.

In the rest of this paper, we first briefly review the concept of LF imaging and the refocusing algorithm in Section II. Then, we describe the conventional and our proposed LF compression framework in Section III. Following, the algorithm of reconstructed focus map and all-in focus image is presented in Section IV. Finally, Sections V and VI present experimental results and conclusion, respectively.

II. Light Field Imaging: A Review

In order to better understand how LF camera captures image, here we briefly review the basic principle of LF imaging and its camera design structure. Different from the conventional camera, an image captured by an LF camera can be refocused. The refocus at different distance or depth is controlled by a parameter of refocusing function, denoted by ρ . A focal stack refers to a set of images generated by a few different ρ values. Very basic of the rendering algorithm of focal stack is described below, but more detail can be found in [15].

1. Structure of light field camera

The plenoptic function is a 7D function $P(x, y, z, \gamma, \theta, \lambda, t)$ which describes the light travelling from the real world to the observer (e.g., camera). It is a function of spatial coordinate (x, y, z) , direction of light ray (γ, θ) , wavelength λ , and time t , all of which allows one to reconstruct image from every possible viewpoint. The function having many parameters can be simplified by assuming static scene (so dropping time t) and a fixed RGB color spaces (so dropping the wavelength λ), so the plenoptic function

can be written $P(x, y, z, \gamma, \theta)$. It is still described in terms of the position and the direction of incoming light to camera sensor. Levoy et al. [16] represented the function as $L(u_1, u_2, s_1, s_2)$ using only four parameters by parameterizing the finite space between two planes. Fig. 1(a) denotes the main lens as the first plane described by coordinate (u_1, u_2) , where micro-lens array (MLA) is the second plane described by coordinate (s_1, s_2) .

The first commercial plenoptic camera (known as Lytro camera) which can capture LF data has installed a MLA before the sensor of the conventional camera [5] as in Fig. 2. A 2D sensor is placed behind the MLA at a distance β , where β is the focal length of the micro-lens. In this structure, the micro-lens separates incoming light rays by directions and the sensor elements of the 2D array behind each micro-lens records the separated lights. Once the raw LF image is captured, one needs to decode the raw LF image, $R(r)$, into the 4D representation of the LF $L(u, s)$, where $r = [r_1, r_2]^T$ is a vector representing a spatial location on the sensor; $u = [u_1, u_2]^T$ and $s = [s_1, s_2]^T$ indicate respectively positions shown in Fig. 1(b). If the LF camera has $M \times N$ micro-lens and $P \times P$ pixels behind each micro-lens, one can extract $P \times P$ views of $I_u(s)$. This process can be done by using the Light Field Matlab Toolbox in [17]. For example, the decoded 4D LF captured by Lytro Illum camera which we use in our paper has 15×15 views, and each view has a size of 635×434 pixels.

2. Refocussing in light field camera

An image captured by LF camera can be used to synthesize an image having different focus, and we will briefly review it here [15]. As seen in Fig. 2, the object is focused by the main lens on the MLA located at a distance x_0 from the main lens, and only the object located at depth z_0 will be perfectly focused according to the well-known relationship given below:

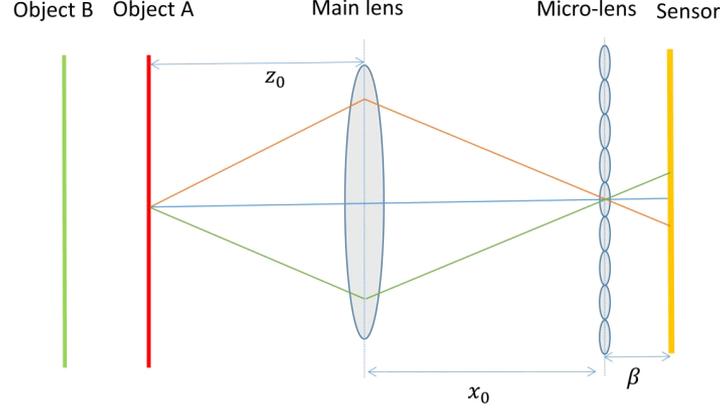


Fig. 2. Structure of LF camera (Lytro Illum camera). The camera captures an image of Object A at distance z_0 with its sensor array placed behind the main lens at x_0 . The micro-lens array is located at β apart from the sensor. The rays of light having different directions (orange, blue and green color) coming from the same point of object are separated by micro-lens and recorded by the sensor behind each micro-lens.

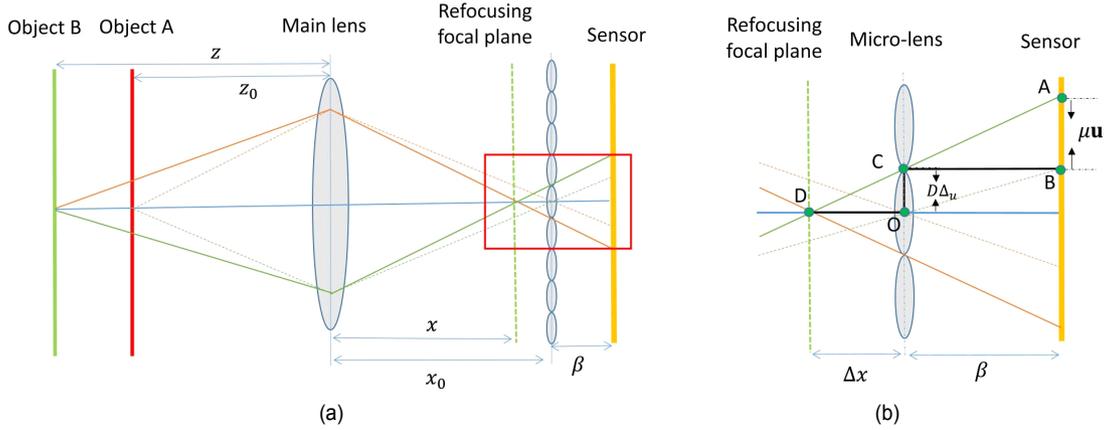


Fig. 3. (a) Light field image refocused from distance z_0 to z . (b) Zoom-out of the red box in (a). The difference between the focal planes before and after refocusing is Δx which can be calculated based on the design parameters of camera

$$\frac{1}{f} = \frac{1}{x_0} + \frac{1}{z_0}, \quad (1)$$

where f is the focal length of the main lens. From equation (1), we see that if the object is at infinity ($z_0 = \infty$), its in-focus image will be formed exactly at the focal length of the main lens ($x_0 = f$), otherwise its in-focus image will be located at a focal plane at x_0 with $x_0 \neq f$. Different from conventional camera, LF camera allows to compute any refocused image (RFIs) at a focal plane x , with $x \neq x_0$. After an image already captured, the image corresponding to a focal plane at x_0 can be computed directly by summing

all the views $I_u(s)$ up to have the RFI $I(s; x_0)$ as:

$$I(s; x_0) = \sum_{\forall u} I_u(s). \quad (2)$$

As can see in Fig. 3(a), the change of focus sets up the refocused focal plane at x , therefore, the summation is done among the shifted views as:

$$I(s; x) = \sum_{\forall u} I_u(s + \Delta_u). \quad (3)$$

As shown in Fig. 3(b), we use a similar triangle rule between two triangles DOC and CBA to compute the shift

amount Δ_u required in equation (3) as below:

$$\frac{DO}{CB} = \frac{OC}{BA} \Leftrightarrow \frac{\Delta x}{\beta} = \frac{D\Delta_u}{\mu u} \Rightarrow \Delta_u = \frac{\mu}{\beta D} \Delta x u, \quad (4)$$

where μ is the pixel pitch and D is the diameter of the micro-lens. Note that the equation (4) depends on μ , β , D , and Δx which is the difference between the original and refocusing focal planes. Now, the refocusing parameter ρ is defined as:

$$\rho = \frac{\mu}{\beta D} \Delta x. \quad (5)$$

Using (4) and (5), (3) can be rewritten as (6) which shows RFI's as a function of the refocusing parameter ρ .

$$I(s;x) = \sum_{\forall u} I_u(s + \rho u). \quad (6)$$

Referring to (1) and noting $x = x_0 - \Delta x$, the focus distance, $z = xf/(x-f)$ can be shown as:

$$Z = \frac{f(x_0 - \Delta x)}{x_0 - \Delta x - f}. \quad (7)$$

Finally, by substituting $\Delta x = \rho\beta D/\mu$ in (5) to (7), one can obtain the following equation:

$$Z = Z_0 \left(\frac{1 - a_0 \rho}{1 - a_1 \rho} \right), \quad (8)$$

where $a_0 = \beta D/\mu x_0$, $a_1 = z_0 a_0/f$, and $z_0 = x_0 f/(x_0 - f)$ is the in-focus distance of the camera.

The model (8) has several properties. Firstly, it is clear that when $\rho=0$, the refocusing distance $z = z_0$, corresponds to a case without performing any refocusing as expected. If the refocusing distance is larger than original (that is, $z > z_0$), $\rho > 0$, and otherwise (that is, $z < z_0$), $\rho < 0$. In this paper, for the Lytro Illum camera, we choose $\rho=1$ which corresponds to a focus distance z at infinity and $\rho=-1$

which corresponds to a focus distance closest next to the camera as in [11].

III. Light Field Image Compression

In this section, we first describe the state-of-the-art method [18] to compress LF image based on the 4D LF representation for general applications. Under the observation that the amount of data to process is still large for application requiring only refocusing, we investigate a new framework designed especially for refocusing applications only. Its details are described as following.

1. Review of state-of-the-art method: 4D LF-based Compression

The 4D LF based compression scheme that we consider in this work is described in Fig. 4. Basically, it has four main processes: format conversion, pseudo sequence conversion, compression, and rendering focal stack. Each process is explained as following.

The raw LF image which is of high dimensional data can have one representation among lenslet, 4D LF, or epipolar formats. Thus, at first, we need to find out the best data format which is supposed to bring the most coding efficiency. The data formats can be also either non-invertible or invertible with respect to the type of camera. In this paper, we consider a conversion from the lenslet data, captured by plenoptic camera 1.0 (i.e., Lytro camera), to 4D LF. It is invertible since one can convert to either way between two kinds of data format without losing any information. The conversion can be done by using the Light Field Matlab Toolbox [17]. I. Viola et al. [18] shows that 4D LF data format gives significantly better results in terms of coding efficiency when compared to compressing data in lenslet or epipolar data formats. Therefore, we select the 4D LF data format as the input of conventional

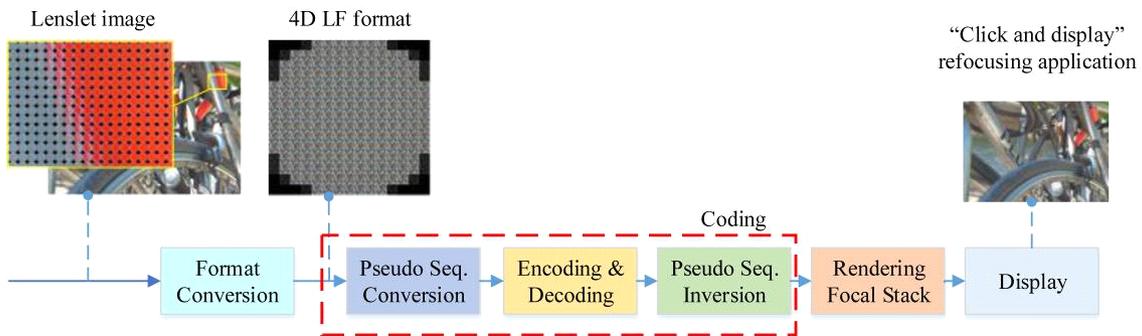


Fig. 4. A state-of-the-art framework for conventional LF image coding: 4D LF based compression

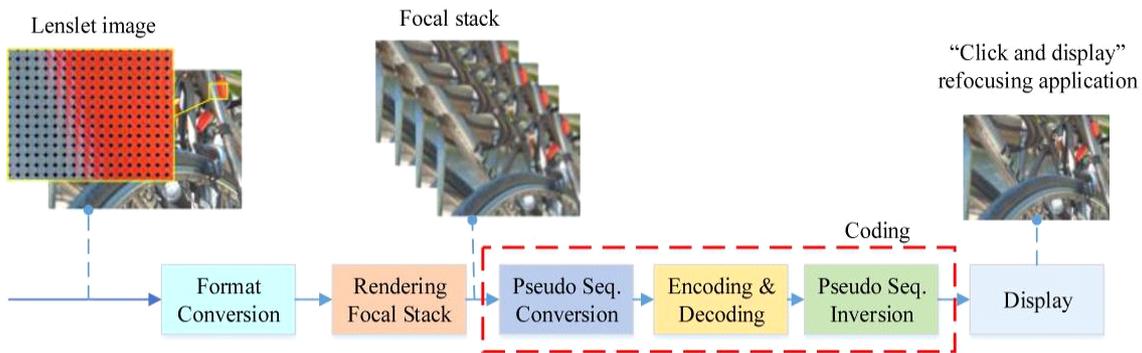


Fig. 5. A proposed framework for LF image coding for refocusing applications: Focal Stack based compression

compression scheme to compare with our proposed framework.

Subsequently, we arrange those sub-aperture images (SAIs) as a pseudo sequence following a selected scan order in such a way that the adjacent SAIs have higher similarity. A scan order can be selected to ensure each SAI is as close as possible to its reference frame, hence, the coding scheme can benefit most from inter-prediction. The work [19] investigated different scan orders such as raster scan, rotation scan, or combination of these scan orders to show that careful selection of the order can bring a significant improvement in compression. In this paper, we choose the raster scan order to form a pseudo sequence same as in [18].

In the next step, the video sequence is coded with video encoders such as the High Efficiency Video Coding (HEVC)

encoder which is the most recent video coding standard. Due to the black views in the corner, we only select 13x13 SAIs to encode. Regarding this problem, authors [20] also tried to select a different subset of SAIs (that is, 9x9, 11x11, 13x13, and 15x15 SAIs) to evaluate the quality of RFI rendered from these subset images. That selection should be of a square shape in order to perform rendering RFI's. The results show that the selection of 13x13 SAIs has a balance between coding efficiency and quality of RFI's. For example, if we only select a small subset of images such as 9x9 SAIs, the out-of-focus region in the RFI still contains sharp edge and has very shallow depth of field so the refocus range (i.e., the maximum extent the image can be refocused) is limited. Finally, after compression, we convert the video sequence to 4D LF format to generate RFI's according to the shifting and adding algorithm in equation (6).

2. Proposed method: Focal stack based compression

As illustrated in Fig. 5, we propose to compress the focal stack instead of 4D LF format in order to reduce the amount of data to process. The proposed scheme consists of four main processes in the same way as the conventional scheme except for the rendering step of the focal stack performed before the coding process. Encoder can reduce redundancy among SAIs in the 4D LF format. But at decoder side, we still need to reconstruct the original 4D LF data again to perform refocusing function. This means that we still need significant storage space. Compared to compression approach using 4D LF format, the conversion of the 4D LF data to the focal stack followed by direct compression extremely reduces the data amount to transmit or store. Moreover, it is not only easy to synthesize the focal stack from LF data or convert it back [14], but the RFI's in focal stack are also easily compressed because they contain mostly low-frequency components in the out-of-focus regions [21].

In the rendering step of the focal stack, we also generate RFI's in the same way as a conventional approach. The number of RFI's in the focal stack varies depending on the refocus range from the closest focus point in the foreground to the furthest one in the background. Scenes with a large depth variation will require more images, while scenes having all objects close together require fewer images. Therefore, the number of images in the focal stack should be not only large enough to cover all depth levels but also, not too many to cause overlaps of the depth of field between the RFI's. In this paper, we investigate many different numbers of images in focal stack for various image contents to see how much it has effects on the quality of refocused image. Before encoding, we also arrange RFI's as a pseudo sequence following the order that corresponds to the RFI's focused from foreground to background of the image. It is noted that the most related works with our proposed method is in

[21] which compresses the RFI's using DWT-based compression for 4D LF data. Since their goal is to reconstruct the best quality of 4D LF, they also compress the residuals of SAI images as well as the RFI's. Hence, ours is different from them. In addition, our proposed framework exploits the most recent video coding standard, HEVC, which makes us expect better compression compared with the existing work [21].

IV. Reconstruction of Focus Map and All-in-focus Image from Focal Stack

The refocusing application enables users not only to “click and refocus” the image but also to see the all-in-focus image. From the focal stack images, we need to estimate a focus map to guide the display process of RFI's process as well as computing the all-in-focus image. Each pixel in focus map has the value corresponding to the index of RFI which gives the best focus. Therefore, when the users click on a region that demands to be in-focus, the focus map helps to identify the corresponding image in the focal stack which has the best in-focus for the selected region. Moreover, by merging all the in-focus regions, one can reconstruct an all-in-focus image. In this paper, we utilize the method in [22] to estimate the focus map. This algorithm includes two steps. First, we detect the regions having strong gradients in each image of the focal stack and then we expand these regions by using the graph cut optimization [23]. Each RFI contains in-focus and out-of-focus regions. The in-focus region is much sharper than the out-of-focus ones. Given a focal stack having K images, for each image I_k in the focal stack, we compute a layer F_k that indicates focus score of the pixels in I_k :

$$F_k(x, y) = \sum_{m=1}^M a_m U(\|\nabla I_k(x, y)\| - m\delta_k), \quad (9)$$

where $U(\cdot)$ is the unit step function, $\nabla I_k(x, y)$ is the image

gradient value of $I_k(x,y)$, and δ_k is the gradient threshold used for I_k . The focus score $F_k(x,y)$ is obtained by comparing the image gradient value with a sequence of increasing threshold $m\delta_k$, weighted with a coefficient α_m . Subsequently, we use the graph cut algorithm in [23] to find the index of image where a pixel has the strongest gradient (i.e., maximum focus score). According to the scheme in [22], we obtain a focus map l such that $l_X := l(x,y)$ gives the index of the focal stack image where the pixel $X=(x,y)$ is in-focus. The focus map l is computed by minimizing an objective function $E(l)$ of the following form:

$$E(l) = \sum_X D(X, l_X) + \sum_{X \sim X'} S(l_X, l_{X'}), \quad (10)$$

Here $D(X, l_X)$ is a data term that represents the cost of assigning the label l_X to the pixel X . The data cost D is determined with respect to the sharpness scores of the pixels as:

$$D(X, l_X) = (\sum_{m=1}^M \alpha_m) - F_{l_X}(x,y), \quad (11)$$

The second term in (10) controls the smoothness of the variation of label among neighboring pixels X and X' (denote as $X \sim X'$). We set the smoothness cost as:

$$S(l_X, l_{X'}) = \begin{cases} 0 & \text{if } l_X = l_{X'} \\ 0.5 + \frac{\log(l_X - l_{X'})}{\log K} & \text{if } l_X \neq l_{X'} \end{cases}, \quad (12)$$

where K is the number of images in the focal stack. The smoothness cost increases with the label difference among neighboring pixel only at a logarithmic rate, in order to avoid the over larger differences along with directions of depth discontinuity. By minimizing the cost function $E(l)$, we can obtain a focus map, which makes it possible to compute an all in-focus image, $AIF(x,y)$, by following equation:

$$AIF(x,y) = \{I_k(x,y), l_X\}, \quad (13)$$

where a focus map $l_X := l(x,y)$ gives the index of the focal stack image, I_k , and the pixel $X=(x,y)$, is in focus.

Table 1. BDBR (%) and BD-PSNR [dB] quality of images in focal stack performance against the 4D LF based compression scheme [18].

Image	comparison	Number of images in focal stack					
		11	17	23	31	41	51
I01	BDBR(%)	-82.79	-81.47	-80.91	-80.57	-79.40	-78.96
	BD-PSNR[dB]	7.73	7.32	7.22	7.17	6.91	6.81
I02	BDBR(%)	-48.11	-44.34	-43.17	-41.90	-38.59	-34.70
	BD-PSNR[dB]	2.71	2.40	2.67	2.28	2.07	1.86
I03	BDBR(%)	-27.38	-21.29	-18.81	-16.27	-12.91	-5.19
	BD-PSNR[dB]	1.32	0.94	1.49	0.73	0.62	0.26
I04	BDBR(%)	-25.92	-16.23	-13.41	-11.50	-4.30	2.48
	BD-PSNR[dB]	0.76	0.28	0.27	0.24	0.03	-0.20
I05	BDBR(%)	-81.84	-81.22	-80.36	-80.66	-78.89	-78.11
	BD-PSNR[dB]	6.88	6.61	6.42	6.44	6.09	5.91
I06	BDBR(%)	-59.94	-57.34	-55.68	-52.22	-48.05	-40.88
	BD-PSNR[dB]	2.32	2.07	1.95	1.79	1.32	1.20
Average	BDBR(%)	-59.94	-57.34	-55.68	-52.22	-48.05	-40.88
	BD-PSNR[dB]	2.32	2.07	1.95	1.79	1.32	1.20

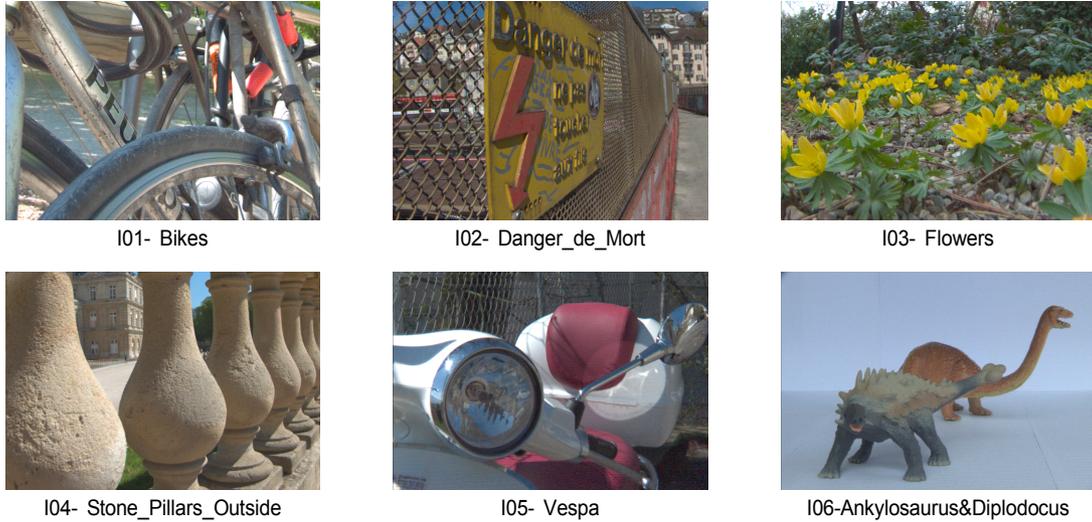


Fig. 6. Six images captured by Lytro Illum camera in the EPFL light field dataset [25].

V. Experimental Results

In this paper, we use a set of six images captured by Lytro Illum camera, namely I01 to I06, in the EPFL light field dataset [25]. They cover many different scenes such as indoor, outdoor, people, and etc. as shown in Fig. 6. The raw lenslet image has a size of 7728x5368 with 10 bits per pixel precision. To obtain 4D LF data, the raw lenslet images were converted by the Light Field Matlab Toolbox [17], where each 4D LF output data contains 225 views (i.e., sub-aperture images (SAIs)) with resolution of 625x434 pixels, 8 bits per pixel precision.

We use the HEVC reference software (HM) version 16.17 [24] for encoding and decoding. QP is set from 10 to 45 to simulate various image qualities (i.e., good to bad quality). The configuration of the encoder is set as follows: GOP structure of I-B-B-B with only the first frame as a reference in order to have the best coding performance [19]. To measure the coding gain of the proposed method, Bjontegaard bitrate (BDBR) and Bjontegaard delta PSNR (BD-PSNR) [26] are measured against the Anchor of the 4D LF based compression approach [18]. A negative BDBR (or positive BD-PSNR) represents better coding ef-

ficiency of the proposed method compared with the anchor.

1. Quality Evaluation: bitrate vs. PSNR

Average \overline{PSNR} is computed by averaging $PSNR$ values of all RFI's in the focal stack, as:

$$\overline{PSNR} = \frac{1}{K} \sum_{k=1}^K PSNR(I_k, I'_k), \quad (14)$$

where I_k is RFI rendered from original LF image and I'_k is RFI rendered from compressed LF image. $PSNR(I_k, I'_k)$ of each RFI in the focal stack is computed as follows:

$$PSNR(I_k, I'_k) = 10 \log_{10} \left(\frac{255^2}{MSE} \right), \quad (15)$$

where the mean squared error (MSE) is defined as:

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_k(i, j) - I'_k(i, j)]^2, \quad (16)$$

where m and n are the dimensions of the RFI (i.e., $m = 625, n = 434$).

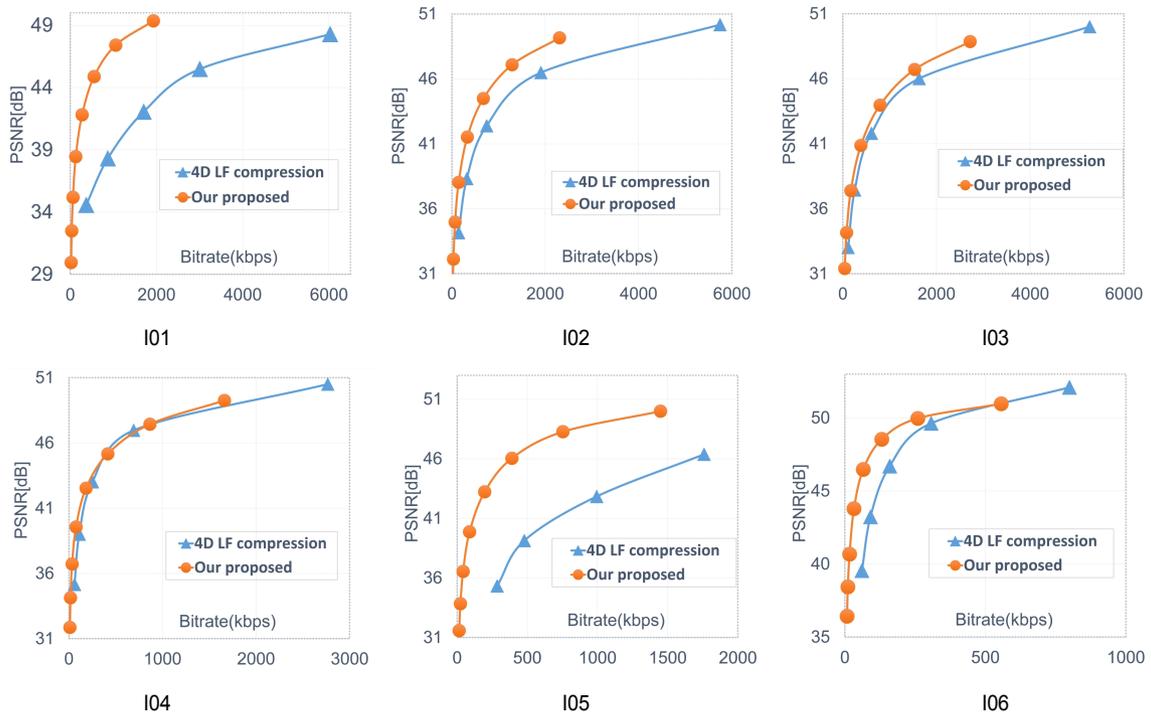


Fig. 7. Coding efficiency comparison of the proposed method with 4D LF based compression in [18] for the Bikes LF images (The number of images in focal stack is $K=11$)

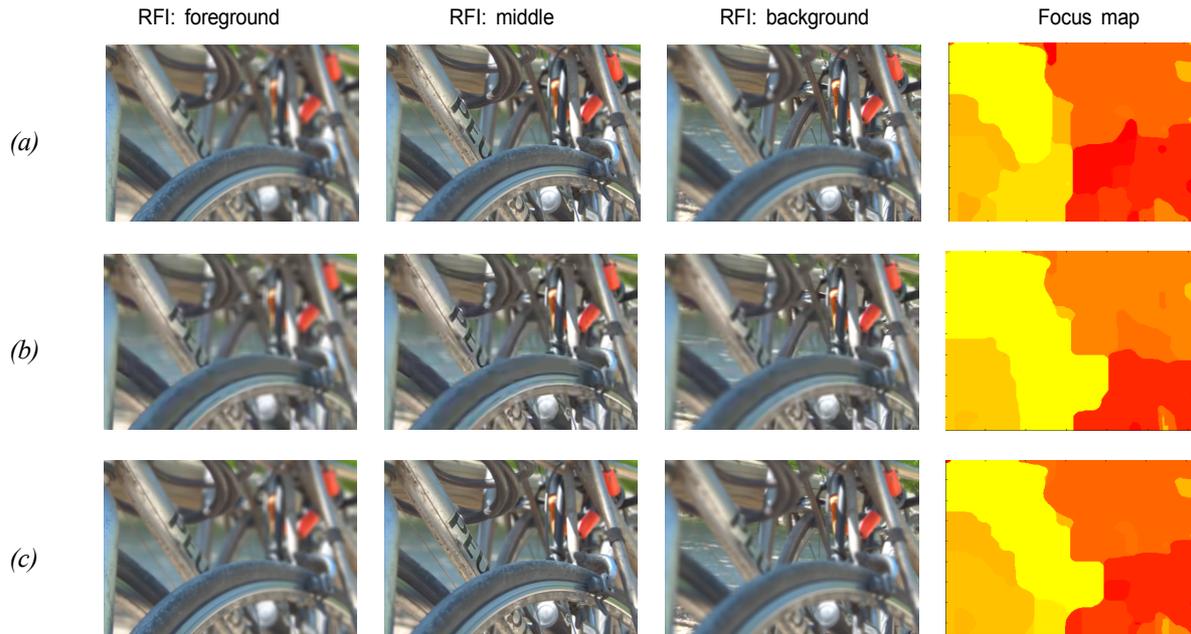


Fig. 8. Quality comparison of RFI for the Bikes LF images: (a) original image; (b) compressed 4D LF image at 872.23Kbps; (c) compressed focal stack image with bitrate at 552.87Kbps



Fig. 9. Quality of all-in-focus images for the Bikes LF images generated from the focal stack in Fig. 8(PSNR [dB]/ SSIM/ bitrate [Kbps])

In order to verify the number of images to generate in each focal stack, we tested several values of $K=11,17,23,31,41,51$. The values of K are chosen to be large enough to fully reconstruct the 4D LF from the focal stack with a good quality [27]. Since our goal is not to reconstruct the original 4D LF, we selected these values just for testing our refocusing applications. As can be seen in Table I, the bitrate decreases when the number of images in focal stack increases. However, the difference is not so large. For example, when $K=11$, BDBR reduces to -59.94% compared to -40.88% when $K=51$. Even with K about five times larger than the smaller one, BDBR did not change much as expected. It is due to the properties of refocused image in which the out-of-focus regions contain much low frequency components which are easy to compress [21].

Fig. 7 shows the rate-distortion performance of the LF images I01 to I06. Since the 4D LF data format is converted to a focal stack data, obviously, our proposed compression scheme has better coding gain compared to the 4D LF based compression scheme, especially at a low bitrate. The BDBR and BD-PSNR depend on the image contents. For example, as can be seen in Fig. 6, I02 and I03 contain a lot of texture regions giving lower bitrate reduction, where I01 and I05 images have many smooth regions which lead to better compress ratio.

2. Quality Evaluation: focal stack vs. all-in-focus images

The refocusing applications allow users to enjoy “click-and refocus” as well as to show the all-in-focus image, which has been shown in [22]. Fig. 8 shows that the first three images on the left illustrate the image with focus on different objects in the image. Focus map has many colors corresponding to different depth layers. Users can click at any layers, and the image belonging to that layers will be displayed on the screen. In [10], authors have verified that the estimated focus map does not have great impact on all-in-focus image creation, but it is only influenced by the distorted texture image in the focal stack. That is why in this paper we did not evaluate the quality of focus map estimated from the focal stack (i.e., focus map can be easily estimated from the focal stack even at a larger QP). In Fig. 9, we can see that our all-in-focus image has much better quality in both PSNR and SSIM because our proposed method still maintains the detailed texture of image compared to the 4D LF compression even at a low bitrate.

VI. Conclusion

In this paper, we proposed a LF compression framework mainly for refocusing applications. We first generate the focal stack and compress it as a pseudo video sequence using HEVC. The results showed that our proposed framework outperforms that of the state-of-the-art 4D LF based compression method in compression performance.

References

- [1] G. Lippmann, "Epreuves reversibles, photographies integrales," *Academie des sciences*, pp. 446-451, 1908.
- [2] H. Ives, "A Camera for Making Parallax Panoragrams," *J. Opt. Soc. Am.*, vol. 17, no. 4, pp. 435-439, 1928.
- [3] M. Levoy, et al., "Light field microscopy," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 924 - 934, Jul. 2006.
- [4] K. Fife, et al., "A 3mpixel multi-aperture image sensor with 0.7um pixels in 0.1um cmos," *IEEE ISSCC Digest of Technical Papers*, pp.48 - 49, 2008.
- [5] R. Ng, et al., "Light field photography with a hand-held plenoptic camera," in *Computer Science Technical Report*, vol. 2, no. 11, pp. 111, 2005.
- [6] ISO/IEC JTC 1/SC29/WG1 JPEG, "JPEG Pleno Call for Proposals on Light Field Coding," in *ISO/IEC JTC 1/SC29/WG1 JPEG, Doc. N74014*, Jan 2017.
- [7] M. Rerabek, et al., "ICME 2016 grand challenge: Light-field image compression," *Call for proposals and evaluation procedure*, 2016.
- [8] I. Viola and T. Ebrahimi, "Quality Assessment of Compression Solutions for ICIP 2017 Grand Challenge on Light Field Image Coding," in *Proc. 9th Workshop on Hot Topics in 3D Multimedia*, July 2018.
- [9] ICME 2018 Grand Challenge on Densely Sampled Light Field Reconstruction, available at <http://www.tut.fi/civit/icme-2018-grand-challenge-densely-sampled-light-field-reconstruction/>.
- [10] M. Rizkallah, et al., "Impact of light field compression on refocused and extended focus images," in *Proc. European Conf. Sig. Process.*, pp. 898 - 902, 2016.
- [11] C. Perra, et al., "Effect of light field subsampling on the quality of experience in refocusing applications," in *Proc. IEEE Int. Conf. Quality Multi. Exper.*, 2018.
- [12] P. Paudyal, et al., "Towards the perceptual quality evaluation of compressed light field images," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 507-522, Sep. 2017.
- [13] P. A. Kara, et al., "Evaluation of the Concept of Dynamic Adaptive Streaming of Light Field Video," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 407-421, June 2018.
- [14] A. Levin and F. Durand, "Linear views synthesis using a dimensionality gap light field prior," in *Proc. IEEE Conf. Comp. Vision Pattern Recog.*, pp. 1831-1838, 2010.
- [15] S. Pertuz, et al., "Focus model for metric depth estimation in standard plenoptic cameras," in *SPRS Journal of Photogrammetry and Remote Sensing*, vol. 144, pp. 38-47, Oct. 2018.
- [16] M. Levoy and P. Hanrahan, "Light field rendering," in *SIGGRAPH*, 1996, pp. 31 - 42.
- [17] D. Dansereau, "Light Field Toolbox v0.4", available at <https://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4>.
- [18] I. Viola, et al., "Comparison and evaluation of light field image coding approaches," *IEEE J. Select. Topic Sig. Process.*, vol. 11, no. 7, pp. 1092 - 1106, 2017.
- [19] T. N. Canh, V. V. Duong, and B. Jeon, "Boundary handling for video based light field coding with a new hybrid scan order," in *Proc. Inter. Workshop on Advanced Image Tech.*, pp. 1-4, Jan. 2019.
- [20] V. V. Duong, T. N. Canh, and B. Jeon, "Light Field Image Coding for Efficient Refocusing," in *Proc. IEEE 12th Inter. Symp. Emb. Multi./Many-core Sys. On Chip*, pp. 74-78, Sept. 2018.
- [21] T. Sakamoto, et al., "A study on efficient compression of multi-focus images for dense light field reconstruction," in *Proc. Vis. Comm. Image Processing(VCIP)*, pp. 1-4, Nov 2016.
- [22] A. Mousnier, et al., "Partial light field tomographic reconstruction from a fixed camera focal stack," available at <http://www.irisa.fr/temics/demos/lightField/LightFieldPartialReconstruction.html>.
- [23] Y. Boykov, O. Veksler, and R. Zabih, "Efficient approximate energy minimization via graph cuts," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1222 - 1239, Nov. 2001.
- [24] HEVC reference software, HM 16.17, available at https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.
- [25] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *Proc. 8th Int. Conf. Quality Multi. Exp.*, pp. 1 - 2, 2016.
- [26] G. Bjontegaard, "Calculation of Average PSNR Differences Between RDCurves," document ITU-T SG16 Q.6, VCEG-M33, Geneva, Switzerland, Apr. 2001.
- [27] F. Perez, et al., "Lightfield recovery from its focal stack," *J. Math. Imag. Vis.*, vol. 56, no. 3, pp. 573 - 590, 2016.

 저 자 소 개



Vinh Van Duong

- 2017 : Hanoi University of Science and Technology (Vietnam)
- 2017 ~ Current : Combined Master & PhD student at Digital Media Lab, Sungkyunkwan University (Korea)
- ORCID : <https://orcid.org/0000-0003-3746-5664>
- Research interest : Video coding, Image enhancement and Light field imaging

저 자 소 개



Thuong Nguyen Canh

- 2012 : Hanoi University of Science and Technology (Vietnam)
- 2014 : Master at Digital Media Lab, Sungkyunkwan Univerisity (Korea)
- 2019 : PhD at Digital Media Lab, Sungkyunkwan Univerisity (Korea)
- 2019 ~ Current : Post-Doctoral Fellow at Institute for Datability Science, Osaka University (Japan)
- ORCID : <https://orcid.org/0000-0002-7880-6659>
- Research interest : Video coding, Compressive sensing, and Light field imaging



Thuc Nguyen Huu

- 2017 : University of Engineering and Technology-Vietnam National University (Vietnam)
- 2017 ~ Current : Combined Master & PhD student at Digital Media Lab, Sungkyunkwan Univerisity (Korea)
- ORCID : <https://orcid.org/0000-0002-4885-4676>
- Research interest : Video coding and Light field imaging



Byeungwoo Jeon

- 1985 : Department of Electronics Engineering, Seoul National University, Seoul, Korea (BS)
- 1987 : Department of Electronics Engineering, Seoul National University, Seoul, Korea (MS)
- 1992 : School of Electrical Engineering, Purdue University, West Lafayette, USA (PhD)
- 1993 ~ 1997 : Signal Processing Lab, Samsung Electronics, Suwon, Korea (Principal Research Engineer)
- 1997 ~ Current : School of Electronic Engineering, Sungkyunkwan University, Suwon, Korea (Professor)
- ORCID : <https://orcid.org/0000-0002-5650-2881>
- Research interest : Multimedia, Video Compression, Signal Processing