Light-Field Raw Data Synthesis From RGB-D Images: Pushing to the Extreme

YIQUN WU,1,2 (Graduate Student Member, IEEE), SHUAICHENG LIU,1 (Associate Member, IEEE), CHAO SUN1, AND BING ZENG1, (Fellow, IEEE)
1School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
2The Hong Kong University of Science and Technology, Hong Kong
Corresponding author: Bing Zeng (eezeng@uestc.edu.cn)

ABSTRACT Light-field raw data captured by a state-of-the-art light-field camera is limited in its spatial and angular resolutions due to the camera’s optical hardware. In this paper, we propose an all-software algorithm to synthesize light-field raw data from a single RGB-D input image, which is driven largely by the need in the research area of light-field data compression. Our synthesis algorithm consists of three key steps: (1) each pixel of the input image is regarded as a spot lighting source that emits directional light rays with an equal strength; (2) the optical path of each directional light ray through the camera’s main lens as well as the corresponding micro lens is considered as accurately as possible; and (3) the occlusion of light rays among objects at different distances within the input image is handled with the depth information. The spatial and angular resolutions of our synthesized light-field data can be scaled up when the input RGB-D image has a higher and higher spatial resolution. Meanwhile, for a given input image with a fixed size, we pay special attention to what would be the extreme we can push the parameters involved in our synthesis algorithm, such as the number of rays emitted from each pixel, the number of micro lenses, and the number of sensors associated with each micro lens. The usefulness of our synthesized data is validated by refocusing, all-in-focus, and sub-aperture reconstructions. In particular, all-in-focus images are evaluated objectively by computing the structural similarity (SSIM) index, which allows us to reach the goal of pushing to the extreme through selecting various parameters mentioned above.

INDEX TERMS Light-field, RGB-D images, micro lens, refocusing, sub-aperture.

I. INTRODUCTION
Light-field is a vector function that records the amount of light flowing in every direction through every point in the 3D space. Such directional light rays can be defined as a 5D plenoptic function [1], parameterized by three coordinates and two angles. If we restrict ourselves to locations outside the convex hull of the object, we can measure the plenoptic function by taking multiple photos using a digital camera, resulting a four-dimensional function.

Light-field cameras can capture three dimensional scene information by recording the directions of the incoming light rays in addition to their intensities. The idea is induced by and similar to using an array of conventional cameras to record different views of a scene at the same time. Clearly, the novelty of a light-field camera is that it has reduced the large volume of an array of cameras or the complicated gantry to just a single shot with a single portable camera. Here, the number of micro lenses determines the spatial resolution of the image and the number of sensors after each micro lens determines the angular resolution of the image. Notice that each sensor produces one pixel in the resulted light-field data.

Today’s light-field cameras available in market are mostly from Lytro and Raytrix. Lytro has developed consumer-level light-field digital cameras that are capable of capturing images using the plenoptic technique; whereas Raytrix is mainly for industrial and scientific applications, with resolutions starting from 1 megapixel at a very expensive price. The newest Lytro model (Lytro ILLUM) can generate raw data that can be extracted by the external light-field toolbox [2], with spatial resolution of $434 \times 625$ and angular resolution...
of $15 \times 15$. There are also some other light-field cameras, such as Pelican imaging [3], Adobe light-field camera [4], CAFADIS camera [5], and Mitsubishi Electric Research Laboratories’s (MERL) light-field camera [6].

Although there are a number of manufacturers working on the production of light-field cameras and many improvements have been made for different applications, there exists a trade-off between the spatial resolution and angular resolution. For the Lytro model mentioned above, though the raw light-field data is with size $6510 \times 9375$ (more than 60 megapixel photosensors totally), its spatial resolution is only $434 \times 625$ and angular resolution is $15 \times 15$. If the photosensor array remains unchanged but we wish to have a higher spatial resolution, the only solution is to reduce the angular resolution. Therefore, with raw data captured by today’s light-field cameras, it seems to be an impossible task if one wishes to reconstruct images (refocusing or all-in-focus) with spatial resolution that can be achieved by today’s high-end conventional cameras (e.g., 4K or 8K) and at the same time to produce a large number of sub-aperture images in order to offer some 3D capability.

We believe that it is still far away to reach the camera-based hardware solution of producing light-field raw data with both sufficiently-high spatial resolution (e.g., 4K or higher) and angular resolution (e.g., $20 \times 20$ or higher). Nevertheless, certain research topics on light-field image data are emerging rapidly. For instance, how to compress light-field image data efficiently has drawn a great attention recently [7]–[11], especially for those with large spatial resolution (4K or higher) and angular resolution (e.g., $20 \times 20$ or higher). This is because that the compression algorithms for high-resolution light-field raw data could be very different from those compression algorithms that are applicable for low-resolution light-field raw data. For instance, the popular video coding standard H.265 [12] defines a lot more intra-prediction modes, as compared to the earlier version H.264 [13], in order to provide a more accurate prediction for each intra frame. According to the ISO/IEC JPEG coding [14], the light-field data compression has been regarded as the focus of the coding of the still image. The evaluation metrics are still under discussion, while they are mainly composed of the objective evaluation metrics (PSNR & SSIM) and the subjective evaluation (animation comparisons). Furthermore, the on-going effort for the next-generation standard H.266 designs even more intra-prediction modes as compared to H.265. One rational for doing this is that the resolution of video frames is becoming bigger and bigger.

In order to develop and test the compression algorithm for high-resolution light-field data, we first need to access the light-field data with high resolution, which is unfortunately impossible with the existing capturing devices. To overcome this dilemma, we are therefore motivated to develop a synthesis-based solution to provide light-field raw image data where the spatial resolution and angular resolution can be made arbitrarily high. For example, we can provide a synthetic light-field raw data with size $80000 \times 80000$ (6400 megapixel totally: $4000 \times 4000$ spatial resolution and $20 \times 20$ angular resolution), which is much higher than the exiting light-field raw data captured by any hardware device.

Specifically, we propose an all-software algorithm to synthesize light-field raw data from a single RGB-D input image to extend the aforementioned resolution limitations existing in the current optical hardware manufacturing. As for the availability of depth information, our algorithm can use the captured depth map (e.g., by Kinect). It can also be applied to the stereoscopic case where the depth information can be estimated from a pair of two images (taken at different viewpoints). Specifically, our synthesis algorithm consists of three key steps: (1) each pixel of the input image is regarded as a spot lighting source that emits directional light rays with an equal strength; (2) the optical path of each directional light ray through the camera’s main lens as well as the corresponding micro lens is considered as accurately as possible; and (3) the occlusion of light rays among objects at different distances within the input image is handled with the depth information.

Given an input image with a fixed resolution, an important goal is to push to the extreme in terms of the resolution (both spatial and angular) of the synthesized light-field raw data. Specifically, this is accomplished by selecting three primary parameters involved in our synthesis algorithm, namely, the number of rays emitted from each pixel of the input image, the number of micro lens, and the number of sensors associated to each micro lens. In principle, the higher these parameters are, the better the rendering result is. However, our study shows that, for an input image with a fixed size, there does exist an upper limit for each of these parameter, which is thus regarded as the extreme as no further improvement can be obtained beyond this extreme. We would also like to point out that our algorithm is scalable to the rise of the input resolution without any re-design cost. With the rise of the input size, our algorithm can still be feasible by increasing the sampling rate of main lens and micro lens. The choice of sampling rate of main lens and micro lens (the spatial and angular resolution) is scalable to the input size. Furthermore, other parameters involved in our synthetic camera system (e.g., focus, aperture, and other camera parameters) are all adjustable according to the specific application of the light-field raw data without limitation of the existing capture devices.

The rest of the paper is organized as follows. In Section II, we review some existing light-field raw data and the related algorithms. In Section III, the imaging process of our synthetic camera system will be illustrated in detail. The evaluation of our synthetic data is presented in Section IV, where both objective and subjective evaluations will be considered and we will pay a special attention to pushing to the extreme through choosing several important parameters involved in our synthetic system. Finally, we summarize our work in Section V.
II. RELATED WORKS

Because of a large demand of light-field raw data with high spatial resolution and high angular resolution in both academic research and practical applications, several algorithms for the synthesis of light-field raw data have emerged in recent years. In this section, we will go through these algorithms and some existing datasets.

A. EXISTING DATASETS

Synthetic light-field data rendered with POV-Ray is provided by MIT media lab [15], with a form of image array. One example is shown in Fig. 1. These light-field data contain either 5 × 5 or 7 × 7 views of different 3D scenes. They are mainly for the research of 3D displays and are rendered on a regular grid that facilitates some easy processing. However, the problem is that they are computer-rendered (with computer-generated model and scene) rather than being captured from real scenes.

The Stanford light-field archive [16] provides light-field data that are captured from real scenes with equipments as camera arrays, where the calibration data is also provided. Since the light-field data is not captured with a light-field camera, the data is again in the form of image array rather than the form of a light-field raw data.

The dataset captured by the first generation of Lytro camera is available in [17]. It contains 30 light-field images both indoor and outdoor. However, these light-field images are with a low resolution (220 × 220 in spatial resolution and 15 × 15 in angular resolution).

Another dataset with an increased resolution (434 × 625 in spatial resolution and 15 × 15 in angular resolution) captured by Lytro ILLUM is available in [18]. It contains 118 light-field images that are classified into ten different categories. As compared to the resolution supported by today’s conventional cameras, the spatial resolution achieved by this Lytro camera is still quite low.

B. EXISTING SYNTHESIS ALGORITHMS

A temporal reconstruction technique has been proposed in [19], which exploits the anisotropy in the temporal light field and permits efficient reuse of samples between pixels. Although its purpose is mainly for the rendering of simultaneous motion blur, depth of field, and soft shadows, it proposes the idea of making use of a temporal light field that is reconstructed by the information of sparse samples to reduce the noise due to the variance of the high-dimensional integrand. Reconstruction of 4D light-fields is proposed in [20], which makes use of sparsity in the Fourier domain. The main contribution is an upsampling of the light-field data by an optimization in the continuous Fourier domain, which solves to a certain extent the problem about the low resolution of light-field data. Meanwhile, how to reconstruct (partially) high-resolution 4D light-fields from a stack of differently-focused photographs taken with a fixed camera is proposed in [21], in which the camera calibration for the stack of photographs has been considered carefully.

On the other hand, some algorithms have been proposed to use the characteristics of light-field data to do the image-based rendering. For instance, image-based rendering from unstructured lumigraph is proposed in [22], a view rendering that conquers the missing of the depth map from several views is proposed in [23], and a novel algorithm is proposed in [24] to simulate the depth-of-field (DOF) effect from the perspective of light-field rendering. The difference between these algorithms and ours is: the goal of the above-mentioned algorithms is to render a new image (new view or new DOF), whereas our algorithm intends to synthesize the high-resolution light-field raw data. Although we also consider several rendering tasks, such as refocusing, all-in-focus, and multi-view, in the verification part of our work, the purpose of doing so is to demonstrate the usefulness and usability of the synthesized light-field data. Because of this, we do not compare our rendering results directly with the most recent state-of-the-art works that often utilize some different settings. However, we believe that the light-field data synthesized in our work offers a rich set of transitional characteristics that would become exploitable in the image-based rendering methods.

Recently, learning method has been proposed in [25] to synthesize the 4D light-field from a single image input. This algorithm works well with one particular type of image (for example, flowers) according to its training dataset, which unfortunately cannot deal with all kinds of images at one time. An End-to-end view synthesis for light field imaging with Pseudo 4DCNN algorithm has been proposed in [26]. In this paper, the main purpose is to find a way to realize the view synthesis with good quality and fast speed, and the algorithm needs a large amount of inputs that need the light-field camera, while the main purpose of ours is to provide a simple and easy-to-execute way (even without a light-field camera) to generate a synthetic light-field raw data that can be applied in various applications, eg. view synthesis, refocusing, depth estimation, compression on LF data and so on. Meanwhile,
we have proposed an algorithm to reconstruct light-field raw data from a single RGB-D image [27]. The present paper provides a big extension to [27] as follows. First, the optical path of each directional light ray through the camera’s main lens as well as the corresponding micro lens is kept in the 3D coordinate; whereas such 3D scenario is simplified into two 2D processes in [27] so that the rendering results are defective visually. Second, all approximations made in [27] are replaced by a much more accurate representation, e.g., sampling of the continuous light-ray emitted from each pixel is now much more dense and all calculations along the optical path of each light ray are now conducted much more precisely. Third, the depth map is kept in its 8-bit coded format, whereas this map has been quantized into 5 layers in [27]. Last, we have tried to push to the extreme through the selection of several important parameters involved in our synthetic system.

III. OUR SYNTHETIC CAMERA SYSTEM

The pipeline of our synthesis algorithm for light-field raw data is shown in Fig. 2. At the input end, the RGB image is first organized into a 3D point cloud according to the associated depth map (D). Each color pixel is assumed to be a spot lighting source to emit continuous light-ray towards the camera’s main lens. We sample this continuous light-ray into a large number of directional light rays. Each directional light ray will propagate through the camera’s main lens and one corresponding micro lens and eventually be recorded on the corresponding sensor in the sensor array. This process is the core part of our synthesis algorithm, which will be described in details in the following.

A. LIGHT SOURCE ASSUMPTIONS

Whether from the sun or other lighting sources, light rays with certain brightness, color and direction always reflect off objects and travel in various directions to produce a continuous light-field. Of the infinite rays in the light field, we can only see those that shine towards our eyes. Conventional cameras follow exactly this mechanism to record the color and intensity of light rays within the camera’s view angle (or aperture).

When we take a picture, the camera records the intensity and color of every light ray reflected from the objects in the scene that the camera is aiming at. When we observe the picture, our eyes perceive the light rays reflected from the picture. Although the reflected light rays are originated from some lighting sources (the sun, some illuminating lights, or a mixture), we can regard that these light rays are emitted from the picture directly. Since the picture is usually a digital one, we make two grounding assumptions as follows:

- Each pixel within the picture serves as a spot lighting source to emit a light-ray continuously along two incident angles.
- After sampling this continuous light-ray, all resulted light rays have the same strength but propagate in different directions.

Note that pixels have different depths so that some directional light rays emitted from a pixel may be blocked by other pixels. Such an occlusion effect will be discussed later in this section.

For scenes containing objects with unknown bidirectional reflectance distribution functions (BRDF), there are two kinds of reflection, i.e., diffuse reflection and specular reflection. For the diffuse reflection, which means that the objects are lambertian, every point of the object can indeed be regarded as a spot light source to emit light rays that are distributed uniformly [28]. For the specular reflection, the situation can be simplified such that only a single directional light ray is emitted from the point. On the other hand, for objects with known BRDF, one can do the corresponding adjustment, such as the angle of each emitting light ray and its strength. Since most objects in the real world are lambertian, the diffuse reflection model has been widely used, which justifies our assumptions.

B. OPTICAL PATH

In this subsection, we will present how to describe as accurately as possible the optical path of each directional light ray...
through the camera’s main lens as well as the corresponding micro lens.

1) SYSTEM SETUP
Our system is made up of four parts: object, main lens, micro lens array, and photosensors. We take a single RGB-D image as the input where the image size is $M \times N$. The parameters of the camera’s main lens as well as each micro lens are listed in Table 1. In general, a lens is composed of two sphere surfaces that are mirror images with respect to each other, as shown in Fig. 3. In practice, we know the object point and the camera. Specifically, the larger the distance to the camera is, the nearer the point is to the camera. We first denote by using $z_{\text{near}}$ and $z_{\text{far}}$, respectively, the distance of the nearest point to the camera (with the biggest value $i_{\text{f}}$ in the 8-bit depth map) and the distance of the farthest point to the camera (with the smallest value $i_{\text{n}}$ in the depth map). Then, for any point $(x, y)$ in the image with depth value $i_{\text{d}}$, its distance to the camera is

$$z = z_{\text{near}} + \frac{z_{\text{far}} - z_{\text{near}}}{i_{\text{d, max}} - i_{\text{d, min}}}(i_{\text{d, max}} - i_{\text{d}})$$

so that the pixel at $(x, y) (x \in [1, M], y \in [1, N])$ in the image with depth value $i_{\text{d}}$ can be transformed to 3D coordinates $(X_{z}, Y_{z}, Z_{z})$ as follows:

$$X_{z} = \frac{H_{z}}{M} \left(1 - \frac{x}{M} \right) - \frac{z_{\text{far}} - z_{\text{near}}}{i_{\text{d, max}} - i_{\text{d, min}}}(i_{\text{d, max}} - i_{\text{d}})$$

$$Y_{z} = \frac{W_{z}}{N} \left(1 - \frac{N}{Y_{z}} \right)$$

$$Z_{z} = -z$$

2) GENERATION OF 3D POINT CLOUD
With an input RGB image and its corresponding depth map, we only have the pixel coordinates $(x, y)$ ($x = 1, 2, \ldots, M$ and $y = 1, 2, \ldots, N$) and its corresponding depth value $i_{\text{d}}$ (usually coded in an 8-bit depth map), without knowing the real position in the world coordinate system. In order to generate the 3D point cloud of the input RGB-D image, we need to calculate the corresponding coordinates $(X_{z}, Y_{z}, Z_{z})$ in the world coordinate system for every pixel at $(x, y)$ with $i_{\text{d}}$, as shown in Fig. 4.

The depth map only reveals the relative distance between the object point and the camera. Specifically, the larger the value $i_{\text{d}}$ is, the nearer the point is to the camera. We first denote by using $z_{\text{near}}$ and $z_{\text{far}}$, respectively, the distance of the nearest point to the camera (with the biggest value $i_{\text{f}}$ in the 8-bit depth map) and the distance of the farthest point to the camera (with the smallest value $i_{\text{n}}$ in the depth map). Then, for any point $(x, y)$ in the image with depth value $i_{\text{d}}$, its distance to the camera is

$$z = z_{\text{near}} + \frac{z_{\text{far}} - z_{\text{near}}}{i_{\text{d, max}} - i_{\text{d, min}}}(i_{\text{d, max}} - i_{\text{d}})$$

so that the pixel at $(x, y) (x \in [1, M], y \in [1, N])$ in the image with depth value $i_{\text{d}}$ can be transformed to 3D coordinates $(X_{z}, Y_{z}, Z_{z})$ as follows:

$$X_{z} = \frac{H_{z}}{M} \left(1 - \frac{x}{M} \right) - \frac{z_{\text{far}} - z_{\text{near}}}{i_{\text{d, max}} - i_{\text{d, min}}}(i_{\text{d, max}} - i_{\text{d}})$$

$$Y_{z} = \frac{W_{z}}{N} \left(1 - \frac{N}{Y_{z}} \right)$$

$$Z_{z} = -z$$

where the original $M \times N$ image will be amplified to $H_{z} \times W_{z}$ at distance $z$ from the lens.

---

**TABLE 1.** Parameters of optical structure of the synthetic camera system and their notations.

| Parameters          | Main lens | Micro lens |
|---------------------|-----------|------------|
| Refraction index    | $\eta_{\text{main}}$ | $\eta_{\text{micro}}$ |
| Radius of sphere    | $R_{\text{main}}$ | $R_{\text{micro}}$ |
| Thickness           | $d_{\text{main}}$ | $d_{\text{micro}}$ |
| Focal length        | $f_{\text{main}}$ | $f_{\text{micro}}$ |
| Diameter            | $D_{\text{main}}$ | $D_{\text{micro}}$ |
3) LIGHT RAY PROPAGATION

Suppose that the pixel value of a point at \((X_c, Y_c, Z_c)\) is \(I\) and the light-ray irradiated from this pixel toward the main lens is sampled into \(K \times K\) directional light rays. This is equivalent to sampling the main lens into \(K \times K\) cells so that each cell receives one light ray. Then, every light ray irradiated by the pixel has intensity of

\[ i = \frac{I}{K \times K} \times C_{\text{area}}, \quad (4) \]

where \(C_{\text{area}} = \frac{K \times K}{K \times K}\) and \(h \times w\) is the real coverage of the reflection of the pixel \(I\). If the pixel is lambertian, \(h \times w = K \times K\), and the intensity of every light ray irradiated by one pixel with value \(I\) is \(i = \frac{I}{K \times K}\), which is the diffuse reflection. If the pixel is regarded as being caused by the specular reflection, the real coverage of the main lens is \(h \times w = 1\) and \(C_{\text{area}}\) should be \(K \times K\), and the resulted \(i = I\).

Here, we only consider the diffuse reflection so that all pixels are lambertian. In addition, we do not consider the decaying of intensity of a light ray during the propagation.

Referring to Fig. 5, a directional light ray \(l_1\) is irradiated from an object point \(P_{\text{start}}\) and arrives at the front surface of the main lens at point \(P_{\text{front}}\). Then, it refracts through the front surface of the main lens and reaches the point \(P_{\text{back}}\) at the back surface of the main lens, with the direction changed to \(l_2\). After refracting through the back surface of the main lens, it changes the direction to \(l_3\) and hits the point \(P_{\text{end}}\) on the front surface of one micro lens in the micro lens array, which is placed at the focal length of the main lens.

Two directional lines \(l_2\) and \(l_3\) as well as their corresponding end points (i.e., \(P_{\text{back}}\) and \(P_{\text{end}}\)) can be calculated through Algorithm 1.

Specifically, the directional line \(l_1\) is known (because both \(P_{\text{start}}\) and the irradiating direction are known) so that \(P_{\text{front}}\) can be located. Steps 1-4 in Algorithm 1 solve the directional line \(l_2\), after which the point \(P_{\text{back}}\) can be determined. Then, \(l_2\) will refract out the back sphere surface of the main lens and travels continuously to reach one micro lens. Steps 1-4 in Algorithm 1 are repeated here to solve the directional line \(l_3\) and the corresponding point \(P_{\text{end}}\).

Notice that the propagation of a light ray through one micro lens can be modeled in the same way as we did above. Here, we need to translate the coordinate of the system with vector \((0, 0, \frac{d_{\text{main}}}{2}, \frac{d_{\text{main}}}{2})\), which means that the origin of the coordinate system is now at the center of the micro lens array.

In summary, for every directional light ray irradiated by an object point \(P_{\text{start}}\) (i.e., \(l_1\)), its optical path through the main lens and the corresponding micro lens can be determined.

**Algorithm 1 Light Ray Propagation Through Convex Lens**

1. \(l_1 = P_{\text{front}} - P_{\text{start}}\); \(l_{\text{front}} = P_{\text{front}} - P_{\text{front}}\); \(P_{\text{front}} = (0, 0, K_{\text{main}} - d_{\text{main}}/2)\) is the centre of the sphere of the front surface of lens
2. According to Snell’s law \([31]\), \(\cos(\theta_1) = \frac{l_1 \cdot l_{\text{front}}}{\|l_1\|\|l_{\text{front}}\|}\); \(\sin(\theta_2) = \frac{\cos(\theta_1)}{n_{\text{main}}}\);
3. \(\text{norm} = l_1 \otimes l_{\text{front}}\) is the normal of the plane where \(l_1\) and \(l_{\text{front}}\) are in.
4. Solve the following equations to get \(l_2\)
   \[
   \begin{align*}
   l_2 \cdot \text{norm} &= 0 \\
   l_2 \cdot l_{\text{front}} &= \cos(\theta_2)\|l_{\text{front}}\|\|l_2\| \\
   l_2 \cdot P_{\text{front}} &= 0
   \end{align*}
   \]
5. \(P_{\text{back}}\) is the intersection point of \(l_2\) and sphere \(X^2 + Y^2 + (Z + (R - d/2))^2 = R^2\) (centred at \(P_{\text{back}} = (0, 0, -(R - d/2))\)).
6. Repeat Step 1 to 4 to get \(l_3\) and \(P_{\text{end}}\).
accurately. Then, the intensity recorded at the corresponding photosensor is $i_s = i \cdot \cos(\theta_{\text{end}})$ where $\theta_{\text{end}}$ is the angle. After adding up intensities of all light rays arrived at the same sensor, we can get the final light-field raw data. The computational complexity is related to the input image size and the sampling of the main lens, i.e., $\mathcal{O}(M \times N \times n_{\text{main}}^2)$.

C. OCCLUSION EFFECT

In order to model our synthetic imaging system as accurately as possible, we need to consider the noise and distortion in the input data and the corresponding depth map generation. Eventually, this would contribute to yield more accurate as well as high accuracy 3D scene information, a refinement process similar to that in [33].

1) PREPROCESSING OF INPUT DATA

First of all, we would like to point out that there are no specific requirements on the input RGB image, i.e., any color image with any resolution is acceptable in our system.

On the other hand, since our synthetic imaging process requires relatively accurate 3D scene information, a refinement process similar to that in [33] is usually recommended, particularly when the input depth map is composed of large part of holes and/or corrupted by noise. To this end, we choose to apply the depth map refinement algorithm proposed in [33] to fill the holes and sharpen the boundary of the object.

Obviously, the more accurately the depth map matches the corresponding color image, the more accurate the 3D point cloud generation will be. Eventually, this would contribute to a more accurate synthesis of light-field raw data. Depth maps with high accuracy can be obtained by high-end depth cameras. Nevertheless, this solution seems quite expensive. On the other hand, today’s depth cameras (e.g., the newest Kinect) can support $1000 \times 1000$ only. Other solutions include designing some algorithms to enhance the accuracy of today’s depth cameras or estimating as accurately as possible the depth map from two or more pictures that are taken at different angles with high-end conventional cameras.

2) OCCLUSION CHECK

After we get the input RGB image and its corresponding refined depth map, we can transfer every pixel to its world coordinates according to Eq. (3). In this 3D point cloud, some light rays emitted from a pixel might be blocked by other objects that are sitting in front, leading to the occlusion effect, as illustrated in Fig. 6. We carry out the occlusion check according to Algorithm 2. If the light ray is indeed blocked, we will neglect its influence and regard that this light ray cannot reach the final sensor plane.

### Algorithm 2 Occlusion Check

1. for $z_{\text{layer}} = z_{\text{near}} \cdot z_{\text{pixel}}$ (where $z_{\text{near}}$ is the nearest depth layer to the camera and $z_{\text{pixel}}$ is the depth layer where the checked light source is at.) do
2. Find $P_{\text{ins}}(X_{\text{ins}}, Y_{\text{ins}}, Z_{\text{ins}})$ - the intersection point of light ray $I$ (irradiated from the light source at depth $z_{\text{pixel}}$) and the plane at $z_{\text{layer}} = Z_{\text{ins}}$;
3. Find $(x_{\text{ins}}, y_{\text{ins}})$ - the integer coordinate of $P_{\text{ins}}$ at depth layer $z_{\text{layer}}$, obtained by rounding $X_{\text{ins}}$ and $Y_{\text{ins}}$ of $P_{\text{ins}}$;
4. Check the depth image at $z_{\text{layer}}$, $Id_{\text{layer}}$, that if the pixel value of $Id_{\text{layer}}$ at $(x_{\text{ins}}, y_{\text{ins}})$ does not equal to 0. If so, exit the loop and the light ray $I$ is assumed to be blocked.
5. end for

IV. EVALUATION AND APPLICATION

The purpose of our algorithm is to synthesize the light-field raw data, but the ground-truth data is not available. Because of this, we conduct the objective evaluation between the all-in-focus image from the synthesized light-field data and the original input RGB image by computing the structural similarity index (SSIM index) [34]. The subjective evaluation is carried out by evaluating the results of several main applications of the light-field data, such as refocusing, all-in-focus and multi-view. One example of our synthetic light-field raw data is shown in Fig. 7. The input RGB-D data is obtained from [32]. In this example, the input RGB image and the corresponding depth map are of size $1000 \times 1000$. However, only a portion of size $400 \times 400$ is shown in Fig. 7(a) and (b). A portion of the synthesized raw data is shown in Fig. 7(b). Visually, the result shown in Fig. 7(c) looks pretty much like a light-field data. Nevertheless, we need to evaluate, both subjectively and objectively, the usefulness of our synthetic light-field raw data. In this section, this job will be done through three reconstructions, i.e., refocusing, all-in-focus, and sub-apertures. In this process, we pay a special attention to pushing to the extreme by selecting the involved parameters, namely, the number of directional light rays emitted from each pixel, the number of

1 Please go to http://pan.baidu.com/s/1c1263W to observe the original raw light-field data. This website also includes the raw light-field data with size $2000 \times 2000 \times 21 \times 21$. 

FIGURE 6. Light ray B emitted from a further layer is blocked by the inner layer, while light ray A will pass through to reach the main lens.
micro lenses, and the number of sensors associated to each micro lens.

A. RENDERING APPLICATIONS

For the refocusing application, which is performed according to [35], we perform the subjective evaluation because each refocused image can hardly be aligned accurately with the original image. On the other hand, all-in-focus images will be evaluated objectively by computing the SSIM index [34] between the all-in-focus image and the original RGB image, where the all-in-focus image is calculated by [36]. Lastly, sub-aperture images will be compared with the multi-view results that are obtained by the depth image-based rendering (DIBR) algorithms [37]–[40].

B. CHOICES OF PARAMETERS

Different choices of parameters in our synthetic camera system will result in different rendering results in terms of subjective and objective evaluations. Suppose that the optical structure of our synthetic light-field camera system is fixed, we will put emphasis on choices of three parameters: the number of light rays (towards the main lens) \( K \times K \), the number of micro lenses \( P \times P \), and the number of sensors for one micro lens \( Q \times Q \). We will demonstrate how to push to the extreme through selecting these parameters.

Since the input depth map (either from the internet database or from Kinect) is limited in its size at \( 1000 \times 1000 \), the size of the input RGB image is set to \( 1000 \times 1000 \). However, we would like to emphasize that this size can be easily extended to a higher level (e.g. 4K or 8K) as long as the depth map of a larger size is available.

1) EXTREME OF SAMPLING OF LIGHT RAYS

The light-ray emitted from a spot lighting source is continuous along two incident angles so that there are an infinite number of light rays. We need to do a sampling of these light rays, resulting \( K \times K \) light rays. In principle, the bigger the value is assigned to \( K \), the better the rendering performance will be, which will be demonstrated in the next subsection through computing the SSIM index between the all-in-focus image and the original input image. However, these results also show that, with the rise of the sampling rate \( K \), the SSIM curve tends to may saturate at some point. Therefore, we can regard such a turning point as the extreme choice of the sampling rate \( K \).

2) EXTREME OF SPATIAL RESOLUTION

The spatial resolution of a light-field data is determined by the number of micro lenses \( P \times P \). In principle, \( P \) can be pushed to an extremely high level. However, the original input image is with a spatial resolution of \( M \times N \). The question is that whether it is necessary to push the micro lens array size beyond the original resolution. The answer will be provided in the next subsection, also through computing the SSIM index between the all-in-focus image and the original input image.

3) EXTREME OF ANGULAR RESOLUTION

The angular resolution of the light-field data is determined by the number of sensors \( Q \times Q \) for every micro lens. In our algorithm, there is no limit on the choice of \( Q \), which means that one can push to the extreme by choosing a large \( Q \), e.g. \( 50 \times 50 \) or \( 100 \times 100 \). However, since the angular resolution
C. RENDERING RESULTS
We carry out experiments on several test images to generate the rendering results of three applications: refocusing, all-in-focus, and sub-apertures.

1) REFOCUSING
Four refocused results at different distances from the camera are shown in Fig. 8(a)-(d) where the involved parameters are listed in Table 2. When we focus at a specific distance $z_{\text{focus}}$, the points with $Z = -z_{\text{focus}}$ become clear while others are blurred. The larger $|Z| - z_{\text{focus}}$ is, the more blurred the point at $(X, Y, Z)$ will be. Part (a) of Fig. 8 is focused at the front part of the scene so that the back part gets blurred. The focus is moving from front to back in Fig. 8(a) to (d). The subjective perception of a clear blurring-shift at different focal distances serves as a positive evaluation for this rendering application of our synthesized light-field raw data.

Compared to our previous results shown in Fig. 8(e)-(h) [27], our present results are obviously much better. In particular, all black lines appeared in Fig. 8(e)-(h) have become invisible. There are several reasons as follows. Firstly, the parameters used in [27] are set at a much lower level, e.g., only $81 \times 81$ light rays ($K = 81$) and $200 \times 200$ micro lenses ($P = 200$) are used; whereas the sensor array is the same ($Q = 20$). Secondly, the propagation of each

### TABLE 2. Parameters Initialization.

| Parameters                        | Notation | Value          |
|-----------------------------------|----------|----------------|
| Input image size                  | $M \times N$ | $1000 \times 1000$ |
| Number of light rays              | $K \times K$ | $1001 \times 1001$ |
| Number of micro lens              | $P \times P$ | $1000 \times 1000$ |
| Number of sensors (per micro lens)| $Q \times Q$ | $20 \times 20$ |
| Radius of main lens sphere        | $R_{\text{main}}$ | $19.2$ cm |
| Diameter of main lens             | $D_{\text{main}}$ | $4.0$ cm |
| Maximum thickness of main lens    | $d_{\text{main}}$ | $0.273$ cm |
| Radius of micro lens sphere       | $R_{\text{micro}}$ | $0.0192$ cm |
| Diameter of micro lens            | $D_{\text{micro}}$ | $0.004$ cm |
| Maximum thickness of micro lens   | $d_{\text{micro}}$ | $0.000273$ cm |
2) ALL-IN-FOCUS
The SSIM index [34] has widely been used to measure the similarity between the processed image and the original one. This index has been proven to be more consistent with human visual perception as compared to other quantitative metrics such as the peak signal-to-noise ratio (PSNR) and mean squared error (MSE). In our experiments, we compute the SSIM index between the original input RGB image and the all-in-focus image (also called the extended depth of field image) that is obtained by the digital photomontage algorithm developed in [36]. As an example, the all-in-focus image generated from the raw data shown in Fig. 7(a) is presented in Fig. 9(b). Again, the parameters listed in Table 2 have been used to obtain this result. Comparing it with the original image shown in Fig. 9(a), we obtain the SSIM index at 0.9236. Meanwhile, the SSIM index between the all-in-focus result from the light-field data generated by [27] (Fig. 9(c)) and the original image (Fig. 9(a)) is 0.5974. For the all-in-focus result from the stack of refocused images generated by [24], the SSIM index between it (Fig. 9(d)) and the original image (Fig. 9(a)) is 0.7414.

Thanks to the SSIM index, we now have a quantitative metric to push to the extreme about the quality of all-in-focus images by testing a range for each of the three important parameters involved in the synthesis process. That is, while keeping all other parameters unchanged, we choose to vary the number of light rays \(K\), or the number of micro lenses \(P\), or the number of sensors \(Q\), respectively. The resulted SSIM curves are shown in Fig. 10. Note that these curves are the averaged results over ten test images.

As shown in Fig. 10(a), the more light rays after the sampling, the better the rendering result is. This is in consistency with the real world situation where the number of directional light ray in the 3D space has been simplified into two 2D scenarios (i.e., \(X-Z\) and \(Y-Z\)) in [27] so that all calculations along the optical path of one light ray through the main lens and the corresponding micro lens will not be accurate. Lastly, the depth map has been quantized into 5 layers in [27], whereas we keep the original 8-bit depth map with no quantization in the current work.

Compared to the results generated by [24] that uses a similar ray tracing method to generate the refocused image with stereo images (shown in Fig. 8(i)-(l)), the boundary artifacts are superiorly removed by our algorithm. Since [24] only considers the refocusing task, the synthetic 4D light field serves as an intermediate result so that other tasks such as building the sub-aperture images may not be supported. On the other hand, our algorithm is applicable for not only refocusing but also other tasks as presented in the following.
light rays irradiated from a spot lighting source is infinite. Therefore, one can push \( K \) to be as high as possible to get a better rendering result. However, Fig. 10(a) also shows that the SSIM curve grows slower and slower with \( K \). If we set the acceptable threshold at 0.9 for SSIM, we find that 800 \( \times \) 800 light rays are sufficient for our test images of size 1000 \( \times \) 1000, which may serve as a simple rule for the selection of the parameter \( K \).

According to Fig. 10(b), for the test images of size 1000 \( \times \) 1000, it seems that the SSIM curve reaches its maximum when about 800 \( \times \) 800 micro lenses are used. We believe that this may serve as the second simple rule for the selection of the parameter \( P \). Obviously, when the size of the input image increases, the parameter \( P \) should increase proportionally. Such a relationship also applies to the selection of \( K \).

Finally, Fig. 10(c) shows that, when more sensors are used, the rendering performance declines but very slightly. Generally speaking, an increase on \( Q \) only makes very little influence to the all-in-focus image. In the meantime, we know that the information provided by the input depth map is limited and not very accurate. We believe that this is one reason why more sensors do not contribute positively to the rendering performance in the all-in-focus application. However, we still recommend that a relatively big sensor array be used, as more sensors can produce more sub-aperture images.

3) MULTI-VIEW SUB-APERTURE IMAGES

The straightforward method to compose a sub-aperture image is to group the pixels in a light-field raw data that are located at the same position in the \( Q \times Q \) sensor array corresponding to each micro lens [35]. As a result, each sub-aperture image will be of size \( P \times P \) and corresponds to one view for the desired sub-region of the main lens. Fig. 11(a) shows one result by following this method, where a large portion becomes black, i.e., no pixels can be recovered. The reason is explained below.

Referring to Fig. 12, we assume that three light rays \( I_1, I_2, \) and \( I_3 \) with same direction arrive at the first sensor behind micro lenses \( L_{micro1}, L_{micro2}, \) and \( L_{micro3} \), which represents the direction of view for the sub-aperture image composed by the first sensor of each micro lens. However, for micro lenses such as \( L_{micro4}, L_{micro5}, \) and \( L_{micro6} \), light rays \( I_4, I_5, \) and \( I_6 \) with the same direction are all out of the bound of the main lens, which means that no light rays can reach the first sensor of micro lenses \( L_{micro4}, L_{micro5}, \) and \( L_{micro6} \).

In order to solve this problem, we need to take advantage of light rays within a range of each desired direction. The range of direction of a light ray is determined by the number of sensors \( Q \).

As illustrated in Fig. 12, for micro lens \( L_{micro4} \), we locate the light ray \( I_{4 substitute} \) whose direction differs slightly from the direction of \( I_4 \). This small difference must be within the range that is covered by \( I_4 \); otherwise such a substitute will not be used. Note that \( I_{4 substitute} \) will not go through the centre of \( L_{micro4} \) but be refracted to reach the second sensor of \( L_{micro4} \), thus resulting in a sensor substitute when composing the corresponding sub-aperture. The same substitute applies to \( L_{micro5}, L_{micro6}, \) and so on, and, obviously, the shift of the sensor’s location becomes larger and larger.

The relationship between the direction of the desired view (represented by the location of the sensor behind each micro lens \( (u_{sensor}, v_{sensor}) \), where \( u_{sensor} \in [1, Q] \) and \( v_{sensor} \in [1, Q] \)), the location of the target micro lens \( (u_{micro}, v_{micro}) \), where \( u_{micro} \in [1, P] \) and \( v_{micro} \in [1, P] \), and the shift of the location of the substituted sensors \( (u_{shift}, v_{shift}) \), where \( u_{shift} \in [1, Q] \) and \( v_{shift} \in [1, Q] \) can be derived as follows (the calculation is the same for \( u \) and \( v \) so that we only represent the calculation of \( u \)):

\[
u_{ds.main} = \frac{f_{main}}{f_{micro}} u_{ds.sensor} = P \cdot \left( \frac{D_{micro}}{Q} \right) \left[ u_{sensor} - \frac{Q}{2} \right] \tag{6}
\]

where \( u_{ds.main} \) is the projection of the light ray that represents the desired view on the main lens along the \( X \) axis and \( u_{ds.sensor} \) is the distance between \( u_{sensor} \) and \( \frac{Q}{2} \). Then, we can define the location of the boundary micro
lens $u_{\text{boundary}}$ as

$$u_{\text{boundary}} = \begin{cases} P - \text{ceil}(\frac{u_{\text{ds, main}}}{D_{\text{micro}}}) + 1; & u_{\text{sensor}} \leq \frac{1}{2} Q \\ \text{ceil}(\frac{u_{\text{ds, main}}}{D_{\text{micro}}}); & u_{\text{sensor}} > \frac{1}{2} Q \end{cases}$$

(7)

where $\text{ceil}()$ returns the smallest integer greater than or equal to a given number. For those micro lenses that are out of the boundary, the shift of the location of the selected sensor $u_{\text{shift}}$ should be

$$u_{\text{shift}} = \text{ceil}(\frac{Q}{P} |u_{\text{micro}} - u_{\text{boundary}}|)$$

(8)

The final location of the selected sensor $(u_{\text{select}}, v_{\text{select}})$ follows Algorithm 3.

We have applied Algorithm 3 to the example considered in Fig. 11 where Part (b) shows the result. Compared with Part (a), we have obtained the full-size sub-aperture image.

Finally, Fig. 13 shows an $8 \times 8$ array of sub-aperture images that represent multi-views of the original scene. Compared with the multi-view images generated by the DIBR algorithm [37] and [38] (middle column), and the novel view from [41] (right column), we believe that our results (left column) are quite comparable. In fact, all methods (ours, DIBR and Zhang’s) do have certain drawbacks. For instance, our algorithm produces some too bright regions at the object borders while the DIBR algorithm leads to more dark pixels (holes) in the area other than the object borders and the Zhang’s algorithm [41] has more distortion when the novel view is further from the input view. Here, we do not attempt to produce better results as compared to the DIBR algorithm; alternately, we just wish to show some results that are comparable, which therefore demonstrates that our synthesized raw data is usable.

4) MORE RESULTS

In this part we show more results from different input sources: some existing RGB-D dataset, captured by Kinect, and stereo image pairs, with different parameters setup.

**Algorithm 3 Selection of Corresponding Sensors for One View**

1: if $u_{\text{sensor}} \leq \frac{1}{2} Q$
2: \hspace{1em} if $u_{\text{micro}} < u_{\text{boundary}}$
3: \hspace{2em} $u_{\text{select}} = u_{\text{sensor}}$
4: \hspace{2em} else
5: \hspace{3em} $u_{\text{select}} = u_{\text{sensor}} + u_{\text{shift}}$
6: \hspace{1em} end
7: \hspace{1em} else
8: \hspace{2em} if $u_{\text{micro}} > u_{\text{boundary}}$
9: \hspace{3em} $u_{\text{select}} = u_{\text{sensor}}$
10: \hspace{2em} else
11: \hspace{3em} $u_{\text{select}} = u_{\text{sensor}} - u_{\text{shift}}$
12: \hspace{2em} end
13: \hspace{1em} end
14: Repeat Step 1 to 13 by substituting $u$ with $v$ to get $v_{\text{select}}$. 
In Fig. 14, the synthesized light-field raw data is with size $800 \times 800 \times 20 \times 20$ ($P = 800$, $Q = 20$, and other parameters are the same as in Table 2). The input RGB-D image pair is from Kinect, which can capture color and depth image at the same time.

In Fig. 15, the synthesized light-field raw data is with size $1000 \times 1000 \times 20 \times 20$ ($P = 800$, $Q = 20$, and other parameters are the same as in Table 2). The input RGB-D image pair is from the dataset provided in [32], which can provide a pair of color and depth image with high quality.

In Fig. 16, the synthesized light-field raw data is with size $2100 \times 2100 \times 20 \times 20$ ($P = 2100$, $Q = 20$, and other parameters are the same as in Table 2). The input RGB-D image pair is from the right view of a stereo camera and its estimated depth map from the stereo image pair.

In Fig. 17, the synthesized light-field raw data is with size $2000 \times 2000 \times 30 \times 30$ ($P = 2000$, $Q = 30$, and other parameters are the same as in Table 2). The input RGB-D image pair is from the dataset provided in [32], which can provide a pair of color and depth image with high quality.
In our experiments, the maximum of parameter choices of $P$ (spatial resolution) and $Q$ (angular resolution) is 2100 and 30, respectively. According to the mechanism of our algorithm and the time consumed for the experiments (up to 8 hours on CPU), we can further push the choice of parameters (such as $P$ and $Q$) even higher if we run the synthesis on GPU. Here, the only concern is the memory demand of the uncompressed raw light-field data, which can be solved by the upgrading of storage devices and corresponding compression method of light-field data.

V. CONCLUSION

In this paper, we have proposed an algorithm for synthesis of light-field raw data with a single RGB-D input image. Our synthetic data can easily extend over any limit on the spatial and angular resolution of light-field data. To this goal, we have proposed to make use of the depth information to build up a 3D point cloud from the input RGB image and follow as accurately as possible the optical path within our synthetic imaging system.

To demonstrate the usefulness of our synthesized light-field raw data, we have considered three rendering application, i.e., refocusing, all-in-focus, and sub-apertures, where both subjective and objective evaluations have been carried out. For a given input RGB-D image, the rendering performance depends on the choices of three parameters in our synthetic camera system, including the number of light rays, the number of micro lenses, and the number of sensors corresponding to each micro lens. In particular, for the all-in-focus scenario, we have used the SSIM index to help us push the rendering performance to the extreme by selecting each of these three parameters separately.

Sub-aperture images have been generated from our synthetic light-field raw data. Although these images can provide similar 3D information as compared to the multi-view images produced by the DIBR methods, we notice that such 3D information still cannot deal very well with the dis-occlusion problem, as the input is only from one view. This remains as one of the issues in our future works.

In summary, our synthesis algorithm has effectively solved the problem associated with the limited spatial and angular resolution of the existing light-field raw data. With these synthetic raw data, we believe that more applications can be developed, e.g., how to compress light-field raw data and how to extract the depth information from light-field raw data.

REFERENCES

[1] E. H. Adelson and J. R. Bergen, The Plenoptic Function and the Elements of Early Vision. Cambridge, MA, USA: MIT Press, 1991.
[2] D. Dansereau. Light Field Toolbox v0.4. Accessed: Feb. 2015. [Online]. Available: https://cn.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4
[3] K. Venkataraman, D. Lelescu, J. Duparré, A. Mcahnon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar, “PiCam: An ultra-thin high performance monolithic camera array,” TOGACM Trans. Graph., vol. 32, no. 6, pp. 1–13, Nov. 2013.
[4] M. G. Keats Jonathon and H. Kris, “Popsci’s how it works: 100 megapixel camera,” PopSci.com. Popular Sci., vol. 272, no. 1, p. 56, Jul. 2009.
[5] U. of La Laguna. Cafadis Camera. Accessed: 2008. [Online]. Available: http://www.cafadis.ull.es/
[6] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, “Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing,” ACM Trans. Graph., vol. 26, no. 3, Jul. 2007.
[7] C. Perra and D. Giusto, “JPEG 2000 compression of unfocused light images based on lenslet array slicing,” in Proc. IEEE Int. Conf. Consum. Electron. (ICCE), Jan. 2017, pp. 27–28.
[8] R. Zhong, S. Wang, B. Cornelis, Y. Zheng, J. Yuan, and A. Manteanu, “Li-optimized linear prediction for light field image compression,” in Proc. Picture Coding Symp. (PCS), 2016, pp. 1–5.
[9] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, “Pseudo-sequence-based light field image compression,” in Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW), Jul. 2016, pp. 1–4.
[10] Y. Li, M. Sjostrom, R. Olsson, and U. Jennehag, “Efficient intra prediction scheme for light field image compression,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2014, pp. 539–543.
[11] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, “Compressive light field photography using overcomplete dictionaries and optimized projections,” TOGACM Trans. Graph., vol. 32, no. 4, p. 1, Jul. 2013.
[12] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[13] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Aug. 2003.
[14] I. J. S. W. C. P. T. Ebrahimi, Jpeg Pleno Call for Proposals on Light Field Coding, document ISO/IEC AWI 21794 (JPEG Pleno), Jan. 2017.
[15] M. M. L. Gordon Wetzstein. Synthetic Light Field Datasets. [Online]. Available: http://web.media.mit.edu/~gordonw/SyntheticLightFields/index.php
[16] S. U. C. G. Laboratory. The Stanford Light Field Archive. Accessed: 2008. [Online]. Available: http://lightin.A.eld.stanford.edu/lfs.html
YIQUN WU (Graduate Student Member, IEEE) received the B.S. degree from the University of Science and Technology of China, Beijing, in 2010, and the M.S. and Ph.D. degrees from Chongqing University, Chongqing, China, in 2013 and 2016, respectively. Her research interests include computer vision and computer graphics.

SHUAICHE N LIIU (Associate Member, IEEE) received the B.E. degree from Sichuan University, Chengdu, China, in 2008, and the M.S. and Ph.D. degrees from the National University of Singapore, Singapore, in 2010 and 2014, respectively. In 2014, he joined the University of Electronic Science and Technology of China, Chengdu, where he is currently an Associate Professor with the School of Information and Communication Engineering, Institute of Image Processing. His research interests include image processing and computer vision.

CHAO SUN received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, China, in 2015 and 2018, respectively. His research interests include image processing and computer vision.

BING ZENG (Fellow, IEEE) received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1991. He worked as a Postdoctoral Fellow with the University of Toronto, from September 1991 to July 1992, and as a Researcher with Concordia University, from August 1992 to January 1993. Then, he joined The Hong Kong University of Science and Technology (HKUST). After 20 years of service in HKUST, he returned to UESTC in Summer 2013, through the 1000-TalentScheme at UESTC. At UESTC, he leads the Institute of Image Processing to work on image and video processing, 3-D and multiview video technology, and visual big data. During his tenure with HKUST and UESTC, he supervised more than 30 master’s and Ph.D. students, a recipient of over 20 research grants, filed eight international patents, and published more than 250 articles. He was elected as an IEEE Fellow, in 2016, for contributions to image and video coding. His three representing works are as follows: one paper on fast block motion estimation, published in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), in 1994, has so far been SCI-cited more than 1000 times (Google-cited more than 2100 times), and currently stands at the seventh position among all papers published in this Transactions; one paper on smart padding for arbitrarily-shaped image blocks, published in the IEEE TCSVT, in 2001, leads to a patent that has been successfully licensed to companies; and one paper on directional discrete cosine transform, published in the IEEE TCSVT, in 2008, a recipient of the 2011 IEEE CSVT Transactions Best Paper Award. He was also a recipient of the Best Paper Award at China-Com three times (Xi’an, in 2009, Beijing, in 2010, and Kunming, in 2012), the Best Associate Editor Award, in 2011, and the 2nd Class Natural Science Award (the first recipient) from the Chinese Ministry of Education, in 2014. He was the General Co-Chair of the IEEE VCIP-2016, Chengdu, China, in November 2016. He is currently on the Editorial Board of the Journal of Visual Communication and Image Representation and serves as the General Co-Chair of PCM-2017. He served as an Associate Editor for the IEEE TCSVT, for eight years.

**References**

[17] E. V. A. Mousnier and C. Guillemot, *Lytro First Generation Dataset*. Accessed: 2015. [Online]. Available: https://www.irisa.fr/temics/demos/lightField/index.html

[18] M. Rerabek and T. Ebrahimi, “New light field image dataset,” in Proc. 8th Int. Conf. Qual. Multimedia Exp. (QoMEX), 2016, pp. 1–2, Paper EPFL-NP-218363.

[19] J. Lehtinen, T. Aila, J. Chen, S. Laine, and F. Durand, “Temporal light field reconstruction for rendering distribution effects,” *TOGACM Trans. Graph.*, vol. 30, no. 4, p. 1, Jul. 2011.

[20] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, “Light field reconstruction using sparsity in the continuous Fourier domain,” *TOGACM Trans. Graph.*, vol. 34, no. 1, pp. 1–13, Dec. 2014.

[21] A. Mousnier, E. Vural, and C. Guillemot, “Partial light field tomographic reconstruction from a fixed-camera focal stack,” *Comput. Res. Repository*, vol. abs/1503.01903, Mar. 2015.

[22] C. Buchler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, “Unstructured lumigraph rendering,” in Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH), 2001, pp. 425–432.

[23] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis, “Depth synthesis and local warps for plausible image-based navigation,” *TOGACM Trans. Graph.*, vol. 32, no. 3, p. 30, Jun. 2013.

[24] X. Yu, R. Wang, and J. Yu, “Real-time depth of field rendering via dynamic light field generation and filtering,” *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2099–2107, Sep. 2010.

[25] P. P. Srinivasan, T. Wang, A. Sreeval, R. Ramamoorthi, and R. Ng, “Learning to synthesize a 4 rgbd light field from a single image,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, no. 5, Oct. 2017, p. 6.

[26] Y. Wang, F. Liu, Z. Wang, G. Hou, Z. Sun, and T. Tan, “End-to-end view synthesis for light field imaging with pseudo 4DCNN,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 333–348.

[27] C. Sun, Y. Wu, and B. Zeng, “Synthesis of light-field raw data from RGB-D images,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Montreal, QC, Canada, Sep. 2015, pp. 1448–1452.

[28] J. H. Lambert, *Photometria: Sive de Mensuris et Gradibus Luminis, Colorem et Vmbrarum*. Germany: Augustae Vindelicorum, sumptibus viduae E. Klett, typis C.P. Detaielsen, 1760.

[29] E. Hecht and A. Zajac, *Optics*. Reading, MA, USA: Addison-Wesley, 1987.

[30] C.-K. Liang and R. Ramamoorthi, “A light transport framework for lenslet light field cameras,” *TOGACM Trans. Graph.*, vol. 32, no. 4, p. 16, Mar. 2015.

[31] K. B. Wolf and G. Krotsch, “Geometry and dynamics in refracting systems,” *Eur. J. Phys.*, vol. 16, no. 1, pp. 14–20, Jan. 1995.

[32] D. Scharstein and C. Pal, “Learning conditional random fields for stereo,” in *Proc. CVPR*, 2007, pp. 1–8. [Online]. Available: http://liphp.uni-irbr.de/db/conf/cvpr/cvpr2007.html#Scharstein07

[33] K. R. Vijayanagar, M. Loghman, and J. Kim, “Real-time refinement of Kinect depth maps using multi-resolution anisotropic diffusion,” *Mobile Netw. Appl.*, vol. 19, no. 3, pp. 414–425, Jun. 2014.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[35] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” *Comput. Sci. Tech. Rep.*, vol. 2, no. 11, pp. 1–11, 2005.

[36] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, “Interactive digital photomontage,” *TOGACM Trans. Graph.*, vol. 23, no. 3, pp. 294–302, Aug. 2004.

[37] I. Carlildom and J. Paciorek, “Planar geometric projections and viewing transformations,” *CSURACM Comput. Surv.*, vol. 10, no. 4, pp. 465–502, Dec. 1978.

[38] L. McMillan, “An image-based approach to three-dimensional computer graphics,” Ph.D. dissertation, Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, USA, 1997.

[39] S. B. Kang and H.-Y. Shum, *A Review of Image-Based Rendering Techniques*. Piscataway, NJ, USA: Institute of Electrical and Electronic Engineers, 2000. [Online]. Available: https://www.microsoft.com/en-us/research/publication/a-review-of-image-based-rendering-techniques/

[40] M. Xi, L.-H. Wang, Q.-Q. Yang, D.-X. Li, and M. Zhang, “Depth-image-based rendering with spatial and temporal texture synthesis for 3DTV,” *J. Image Video Process.*, vol. 2013, no. 1, p. 9, Dec. 2013, doi: 10.1186/1687-5281-2013-9.

[41] Z. Zhang, Y. Liu, and Q. Dai, “Light field from micro-baseline image pair,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3800–3809.