High-Quality Multi-View Image Extraction from a Light Field Camera Considering Its Physical Pixel Pixel Arrangement

Shu FUJITA†a, Keita TAKAHASHI†b, Members, and Toshiaki FUJII†c, Senior Member

SUMMARY We propose a method for extracting multi-view images from a light field (plenoptic) camera that accurately handles the physical pixel arrangement of this camera. We use a Lytro Illum camera to obtain 4D light field data (a set of multi-viewpoint images) through a micro-lens array. The light field data are multiplexed on a single image sensor, and thus, the data is first demultiplexed into a set of multi-viewpoint (sub-aperture) images. However, the demultiplexing process usually includes interpolation of the original data such as demosaicing for a color filter array and pixel resampling for the hexagonal pixel arrangement of the original sub-aperture images. If this interpolation is performed, some information is added or lost to/from the original data. In contrast, we preserve the original data as faithfully as possible, and use them directly for the super resolution reconstruction, where the super-resolved image and the corresponding depth map are alternatively refined. We experimentally demonstrate the effectiveness of our method in resolution enhancement through comparisons with Light Field Toolbox and Lytro Desktop Application. Moreover, we also mention another type of light field cameras, a Raytrix camera, and describe how it can be handled to extract high-quality multi-view images.

key words: light field camera, super resolution image synthesis, multi-view image

1. Introduction

Light field, or plenoptic cameras, can capture 3D image information with a single device [1]–[8] and have been attracting attention recently. The most typical configuration of a light field camera has a micro-lens array inserted between the main lens and the image sensor. By the working of the micro-lenses, a dense multi-view image data (light-field data) is multiplexed and imaged on the image sensor. In other words, it is able to capture 3D information in a single image. We can obtain the multi-view image data by demultiplexing the raw captured data (RAW image). Exploiting this multi-view information enables us to achieve various applications such as digital refocusing, depth estimation, and generation of free-viewpoint images [3], [9]–[15].

This paper is mainly focused on one type of light field camera, Lytro Illum [7], but we briefly mention another type of field camera, Raytrix [8], in the final section.

Limitations on the resolution of the multi-view images are unavoidable with cameras such as the Lytro Illum. This problem results from recording multi-view image data with a single image sensor, and thus, the resolution of each viewpoint is limited. To give specific values, the image sensor used in the Lytro Illum has a resolution of 7728 × 5368 pixels, but the image resolution for each viewpoint is no more than 626 × 433 pixels when we use the standard software, Light Field Toolbox [16], to demultiplex a RAW image.

The resolution of each viewpoint image could be increased by aligning the multi-viewpoint images with each other and applying super resolution image synthesis [9], [10], [13], [17], [18]. However, conventional methods have difficulty for handling the particular pixel arrangement in the RAW image used by the Lytro Illum. Each pixel in the RAW image has only one of RGB color components due to a color filter array, and the micro-lenses are arranged in a hexagonal lattice. Conventional methods use demosaicing to reconstruct the color information, and pixel resampling is used to virtually align the lens arrangement with a square lattice. These processes involve data interpolation such as weighted sum, and such interpolation consequently causes loss of the original information in the RAW image, which prevents the effect of super resolution image synthesis.

In this article, we propose a super resolution image synthesis method that handles the physical pixel arrangement used by the Lytro Illum as accurately as possible. The proposed method preserves the original pixel arrangement in the RAW image when demultiplexing it into multi-view images. We do not perform demosaicing, pixel resampling or other interpolation on the original data, similarly to [14]. Specifically, pixel values should be read from the RAW image at sub-pixels because the image sensor and lens array are aligned with sub-pixel accuracy. The proposed method reads pixel values by rounding location coordinates to integer values (equivalent to nearest neighbor interpolation) to avoid performing interpolation on the original pixel value data. Then, the proposed method performs super resolution image synthesis using the multi-view images that maintains the original information in the RAW image. Specifically, to support multi-view images described above, we consider the pixel arrangement determined by the RAW image to construct the observation model for super resolution image synthesis and depth estimation. Through experiments with real images, we show that the proposed method can generate high-resolution images that are clearer than those super-resolved from the output of the standard Light Field Toolbox [16], or refocused images with enhanced resolution created using the software from [19]. We also show that the method achieves clarity equivalent to the all-in-focus images.
produced by Lytro Desktop [20].

Sabater et al. [14] introduced a demultiplexing scheme for maintaining the RAW image information from Lytro Illum like cameras, but their scheme did not include super resolution image synthesis. Our method uses the same demultiplexing method as [14], but to the best of the authors’ knowledge, we were the first to show the effectiveness of the demultiplexing method for super resolution image synthesis. The algorithm used in the proposed method is categorized as standard reconstructive super resolution image synthesis, but we incorporated the specific pixel arrangement originated from the RAW image into the observation model. This paper was developed from preliminary papers [21]–[23], which was focused on Lytro Illum like cameras. In this paper, we also mention another type of light field camera, Raytrix, which has a slightly different configuration from a Lytro Illum.

This paper is structured as follows. Section 2 describes the demultiplexing process in more detail, which is the process for generating sub-aperture images from the RAW image of a Lytro Illum. Section 3 describes the super resolution image synthesis procedure using the sub-aperture images. Pixel-wise registration among the sub-aperture images is necessary for super resolution image synthesis, so this method also includes a procedure for depth estimation. Section 4 describes experiments showing the effectiveness of the proposed method. Section 5 mentions how to extract high-quality multi-view images from another type of light field camera, Raytrix. Section 6 concludes the paper.

2. Generating Sub-Aperture Images

An overhead schematic view of a light field camera is shown in Fig. 1. In the light field camera, a micro-lens array is positioned where the image sensor of an ordinary camera would be, and the image sensor is placed behind it. Light rays arriving at each micro-lens reach the image sensor at different positions, depending on their incident directions. This enables the light-field camera to record both the position and the angle of light rays at the same time. An actual RAW image taken by the Lytro Illum and an enlarged section are shown in Fig. 2. From the enlarged section, we can see the hexagonal lattice of the micro-lens array. A set of pixels behind each micro-lens is called a sub-image. Each sub-image records the light rays reaching from various directions at the position of the corresponding micro-lens. This indicates that relative coordinates of pixels in a sub-image correspond to incident directions of light rays. Therefore, 4D light field information, which is the planar position of the micro-lenses (2D) and the incident direction on that plane (2D), is recorded in the RAW image of the Lytro Illum. Here, we represent the sub-image corresponding to each micro-lens as \((p_h, p_v)\). We also represent the relative coordinates within a sub-image with \((\theta_h, \theta_v)\). Using these variables, each light ray is represented by the combination \((p_h, p_v, \theta_h, \theta_v)\).

The RAW image is regarded as a multiplexed data of dense multi-view images called sub-aperture images. A sub-aperture image consists of the pixels with the same relative coordinates \((\theta_h, \theta_v)\), selected from all sub-images. Letting \(I_{R}(\alpha, \beta)\) be a RAW image, we can represent a sub-aperture image corresponding to \((\theta_h, \theta_v)\) as follows.

\[
I_{\theta_h, \theta_v}(p_h, p_v) = I_{R}(\bar{\alpha}(p_h, p_v) + \theta_h, \bar{\beta}(p_h, p_v) + \theta_v)
\]

\[
(1 \leq p_h \leq l_h, 1 \leq p_v \leq l_v).
\]

Here, \((\bar{\alpha}(p_h, p_v), \bar{\beta}(p_h, p_v))\) denotes the coordinate in the RAW image corresponding to the center of sub-image \((p_h, p_v)\). Symbols \(l_h\) and \(l_v\) are the numbers of horizontal and
vertical micro-lenses, respectively. The sub-image position \((p_h, p_v)\) corresponds to the pixel position in the sub-aperture image, and the relative coordinates \((\theta_h, \theta_v)\) in the sub-image correspond to the viewpoint position of the sub-aperture image. As mentioned earlier, the relative coordinates \((\theta_h, \theta_v)\) correspond to the angle of light rays incident on the micro-lens. As a result, by back-tracing the light rays incident on the micro-lens array with the same angle, we can draw a sub-aperture image as the one whose viewpoint is located where those light rays converge, as shown in Fig. 1. For example, gathering together the upper-left pixels in each sub-image produces the sub-aperture image from a viewpoint in the lower-right.\(^1\) Repeating the above process for each relative coordinate \((\theta_h, \theta_v)\) in the sub-images produces multi-view or sub-aperture images from various viewpoints. The number of sub-aperture images (viewpoints) corresponds to the number of combinations \((\theta_h, \theta_v)\) and is at most the number of pixels in a sub-image. The number of pixels in each sub-aperture image, \(n_h \times n_v\) \((n_h\) and \(n_v\) are respectively the number of horizontal and vertical pixels), is equivalent to the number of micro-lenses, and thus \(n_h = l_h\) and \(n_v = l_v\) in the formation of Eq. (1).

There are three important points in generating the sub-aperture images as described above. The first is that due to a color filter array placed over the image sensor, each pixel in the RAW image includes only one color value: R, G, or B. The second is that the RAW image coordinates corresponding to the integer relative coordinates \((\theta_h, \theta_v)\) in the sub-image are generally floating point values. This is determined by the positional relationship between image sensor and micro-lens array. The third is that the pixel arrangement of the sub-aperture images \(I_{\theta_h,\theta_v}(p_h, p_v)\) is in principle the same as that of the micro-lens array. The micro-lens array is actually a hexagonal lattice, so the pixel arrangement of a sub-aperture image is also a hexagonal lattice and not a square grid.

Generally, sub-aperture images are generated as described below [3], [11], [12], [15], [16], [24]–[26]. For the first issue regarding the color filter array, demosaicing is generally used to generate full RGB color information for each pixel. That is, before applying Eq. (1), demosaicing is applied to the RAW image to generate full RGB color information for each pixel in \(I_R\). For the second issue, the image data is interpolated smoothly to extract pixel values at sub-pixels. Thus, in Eq. (1), pixel values are synthesized by smoothly interpolating the values of neighboring pixels when accessing sub-pixels in \(I_R\). For the third issue regarding pixel arrangement, the pixels of the sub-aperture image are resampled to convert the pixel arrangement into a square-grid. Thus, the sub-aperture images obtained using Eq. (1) are resampled into a \((p_h, p_v)\) square grid with scaling and smooth interpolation. This process can be regarded as virtually converting the hexagonal micro-lens arrangement to a square one. In these processes of demosaicing and resampling, the original RAW image data is interpolated using computational operations such as weighted sum. We thus call such methods interpolation-based methods. Light Field Toolbox [16], which is the standard software published for Lytro Illum like cameras, also uses interpolation-based methods. An example of a sub-aperture image generated by this software is shown in the left of Fig. 3.

Interpolation-based methods are widely used, but they have two issues. The first is that during the process of interpolation, unnecessary information is added while the original information could be lost. The second is that images information from different viewpoints could be mixed when conventional demosaicing is applied to the RAW image of a Lytro Illum. While demosaicing, color information is generally interpolated from neighboring pixels. However, neighboring pixels in a sub-image correspond to different viewpoints because sub-aperture images are multiplexed in the RAW image. In fact, image information from different viewpoints should not be mixed as reported in [14].

In contrast, we generate sub-aperture images maintaining the pixel arrangement based on the RAW image, in the same way as in [14]. In this paper, we refer to this as the direct method. A diagram of the direct method is shown in Fig. 4. Here, the color of each pixel corresponds with the color of the color filter array. Since the RAW image information is maintained, no interpolation of the original data is done. More specifically, each pixel in the sub-aperture image has one color information at most since the pixels are extracted without demosaicing. The position coordinates on the RAW image corresponding to the integer relative coor-

---

\(^1\)An image is flipped vertically and horizontally on the image sensor. By reversing this inversion, a pixel located on top-left corresponds to a top-left viewpoint.
coordinates \((\theta_h, \theta_v)\) are generally floating-point coordinates, but the nearest neighboring pixels are selected by rounding the coordinates to avoid data interpolation. Thus, the following equation is used instead of Eq. (1) to generate the sub-aperture images:

\[
\bar{I}_{\theta_h, \theta_v}(p_h, p_v) = I_R(r(\bar{\theta}_h(p_h, p_v) + \theta_h), r(\bar{\theta}_v(p_h, p_v) + \theta_v))
\]

\((1 \leq p_h \leq l_h, 1 \leq p_v \leq l_v)\). \hspace{1cm} (2)

Symbol \(r\) is a function that rounds up or down to the nearest integer. Furthermore, to maintain the pixel arrangement corresponding to the hexagonal lattice on the micro-lens array, empty pixels are inserted at every two pixels on each line in the sub-aperture images. Thus, the horizontal resolution of sub-aperture images produced by the direct method is double the earlier method, satisfying \(n_h = 2l_h\) and \(n_v = l_v\). The sub-aperture image \(I_{\theta_h, \theta_v}(p_h, p_v)\) can finally be represented as:

\[
I_{\theta_h, \theta_v}(p_h, p_v) = \begin{cases} 
\bar{I}_{\theta_h, \theta_v}(\lfloor p_h / 2 \rfloor, p_v) & \text{if } p \text{ is odd} \\
0 & \text{if } p \text{ is even} 
\end{cases}
\]

\((1 \leq p_h \leq 2l_h, 1 \leq p_v \leq l_v)\). \hspace{1cm} (3)

Here \(p = p_h + p_v\), and \([\cdot]\) represents the floor function. We show an example of a sub-aperture image generated by the direct method in the right of Fig. 3.

The processes for generating sub-aperture images with interpolation and direct methods are shown in Fig. 5. The direct method does not perform demosaicing or resampling, so neither unnecessary information is added to the RAW image, nor original information in the RAW image is lost. Sabater et al. [14] showed that these features are useful for depth estimation. In this paper, we show that they provide great benefits also for super resolution image synthesis.

### 3. Increased Resolution through Reconstructive Super Resolution Image Synthesis

This section describes our method for increasing resolution by registering sub-aperture images from various viewpoints and applying reconstructive super resolution image synthesis to them. As shown in Fig. 1, the viewpoint of a sub-aperture image corresponds with the relative coordinates \((\theta_h, \theta_v)\) in the sub-images. Since the \((\theta_h, \theta_v)\) coordinate system is a square lattice, the viewpoint positions of the sub-aperture images can also be considered as being arranged in a square lattice. Thus, the set of sub-aperture images can be regarded as multi-view images taken by cameras placed at this square lattice. Our method uses a standard reconstructive super resolution image synthesis algorithm, but to make this paper self-contained, we describe it fully and concretely here. The original aspect of this research is that we can perform super resolution image synthesis while accurately handling the pixel arrangement in the RAW image of a Lytro Illum. Specifically, we consider the pixel arrangement determined by the RAW image to construct the observation model for super resolution image synthesis and depth estimation. In explaining the principles of the algorithms below, we use matrices and vectors, but they are implemented using image processing operations to save memory in the actual software. Refer to the appendix for details on how to implement the matrix and vector operations as image processing.

The sub-aperture images used for input are denoted by \(n\)-element one-dimensional vectors \(y^{(k)} \in \mathbb{R}^n\), which include RGB color information of all pixel positions. Here, \(n = n_h \times n_v \times 3\), where multiplication by 3 means that the vector contains 3-channels (RGB) color information. Note here that \(n_h = l_h\) and \(n_v = l_v\) for the interpolation-based method while \(n_h = 2l_h\) and \(n_v = l_v\) for the direct method. Superscript \(k\) is the viewpoint index, and \(k \in \mathcal{K} = \{1, \ldots, K\}\), where \(K\) is the number of sub-aperture images. For the interpolation-based methods, all elements of \(y^{(k)}\) have values, but for the direct method, some of the elements are empty. The proposed method can handle both types of sub-aperture images within the same framework.

For our method, a single viewpoint \(k_c \in \mathcal{K}\) is selected as the reference, and super resolution image synthesis is applied to the sub-aperture image \(y^{(k_c)}\) at the viewpoint \(k_c\). Specifically, a high-resolution depth map \(\mathbf{d} \in \mathbb{R}^{N}\) at the viewpoint \(k_c\) is estimated, multiple sub-aperture images \(y^{(k)}\) are registered to the reference viewpoint based on the depth information, and then the high-resolution image \(x \in \mathbb{R}^N\) is estimated using reconstructive super resolution image syn-

---

**Fig. 4** Schematic diagram of direct method.

**Fig. 5** Process for generating sub-aperture images.
thesis. Here, \( N_d = N_h \times N_v \) and \( N = N_h \times N_v \times 3 \), where \( N_h > n_h \) and \( N_v > n_v \) are respectively the number of horizontal and vertical pixels in the high-resolution image \( x \) that includes RGB color information. By repeating these processes, the high-resolution image and the depth map are alternately estimated. Since their estimation accuracies depend on each other, alternately estimating \( x \) and \( d \) can contribute to increasing their accuracies.

In the remainder of this section, Sect. 3.1 describes how the depth map is estimated from the input sub-aperture images. Section 3.2 describes reconstructive super resolution image synthesis that can increase the resolution by registering the multi-view images each other. Section 3.3 presents a method that alternately estimates a depth map and high-resolution image combining the depth estimation method in Sect. 3.1 and the reconstructive super resolution image synthesis in Sect. 3.2. Section 3.4 describes the pixel arrangement matrix used to accurately handle RAW images of the Lytro Illum when estimating depth and applying reconstructive super resolution image synthesis.

3.1 Estimating the Depth Map

The basic principle for depth estimation is the same as multi-view stereo based on block matching. Given a high-resolution image \( x \) from a given viewpoint, Eq. (4) can be solved to estimate depth map \( d \) with the same viewpoint and resolution as \( x \):

\[
d = \arg\min_d E_{\text{depth}}(x, d),
\]

where the energy function \( E_{\text{depth}}(x, d) \) is defined as:

\[
E_{\text{depth}}(x, d) = \sum_{k \in \mathcal{K}} \sum_{i \in [1, N_v]} C^{(k, i)}(x, d_i),
\]

where \( d_i \) represents the \( i \)-th element of vector \( d \); in other words, the depth value at the \( i \)-th pixel position in the depth map. To ensure that the depth values \( d_i \) cover the range of depths in the object space, they are determined beforehand. Function \( C^{(k, i)}(x, d) \) is the matching cost for the \( k \)-th sub-aperture image \( y^{(k)} \) given a depth value \( d \) at the \( i \)-th pixel of the reference image \( x \):

\[
C^{(k, i)}(x, d) = \|W_i \delta^{(k)}(x, d)\|^2,
\]

where \( \delta^{(k)}(x, d) \in \mathbb{R}^N \) is a vector representing the difference image between \( x \) and \( y^{(k)} \), which includes RGB color information in all pixel positions, given a fixed depth value \( d \). Matrix \( W_i \in \mathbb{R}^{N_v \times N_v} \) is a diagonal matrix, and an operator that leaves only the vector elements corresponding to RGB color values at the the \( i \)-th pixel and the neighbors of that pixel. Equation (6) has the effect of taking the average of all RGB color values in the window area centered on the \( i \)-th pixel. Vector \( \delta^{(k)}(x, d) \) is computed using the following equation:

\[
\delta^{(k)}(x, d) = M^{(k)}(d)U y^{(k)} - P^{(k)}_{N \times N}B x.
\]

In the first term on the right side, \( U \in \mathbb{R}^{N \times r} \) represents an up-sampling operator with nearest-neighbor interpolation. Matrix \( M^{(k)}(d) \in \mathbb{R}^{N \times N} \) is an operator that uniformly translates the pixels according to depth values \( d \). After up-sampling a sub-aperture image \( y^{(k)} \) using matrix \( U \), it is registered to the coordinate system of high resolution image \( x \) by translating it with matrix \( M^{(k)}(d) \). In the second term on the right side, \( B \in \mathbb{R}^{N \times N} \) is a matrix representing point spread function (PSF). The shape of the PSF is assumed to be a uniform function with rectangular support, of which size is \( N_h/n_h \times N_v/n_v \). Here, each color channel is separately handled in the matrices but is processed with the same operation. Matrix \( P^{(k)}_{N \times N} \) is a pixel arrangement matrix that masks pixels according to the structure of \( M^{(k)}(d)U y^{(k)} \). This is described in detail in Sect. 3.4. By taking the difference of the first and second terms, the matching error with the assumed depth \( d \) is computed on the same coordinate system and resolution in the high resolution image. As shown in Eqs. (5) and (6), the square of this matching error is aggregated within the window, and the sum of them over all sub-aperture images is the matching cost for each pixel in the high resolution image. By repeating this process while changing depth \( d \), a cost volume (with matching costs for each pixel and various depths) is composed. The cost volume has the same coordinate system as the high-resolution image. Therefore, minimizing Eq. (4) is a process finding the value of \( d \) that minimizes the cost for each pixel in the high resolution image, and consequently, a depth map \( d \) with the same resolution as the high-resolution image is estimated. In addition, all matrices used in Eq. (7) can also be implemented as image processing operations. This indicates that there is no need to keep very large matrices in the implementation. To eliminate noise, a median filter is applied to the estimated depth map at the final step.

There are a number of issues in the depth estimation method described here. Firstly, there is a potential to cause errors in the process of upsampling with matrix \( U \). Secondly, we have given no consideration to occlusions between viewpoints. We have also not considered spatial continuity of depth when independently optimizing depth values for each pixel in the cost volume. There is room for improvement of these in future research, but for the purpose of increasing resolution in this paper, the depth estimation method described here produced adequate results.

3.2 Reconstructive Super Resolution Image Synthesis

This section describes the method for generating the desired high-resolution image \( x \in \mathbb{R}^N \) \((N = N_h \times N_v \times 3)\) using reconstructive super resolution image synthesis, given the high-resolution depth map \( d \) for a given viewpoint, by registering \( y^{(k)} \in \mathbb{R}^n \) \((n = n_h \times n_v \times 3)\) to that viewpoint. The energy function to be minimized is defined as follows.

\[
E_{\text{SRP}}(x, d) = \frac{1}{2} \sum_{k \in \mathcal{K}} \|y^{(k)} - A^{(k)}(d) x\|^2 + \lambda R(x).
\]

The first term is computed from the observation model for
the sub-aperture images. \( R(x) \) in the second term is a normalization to enhance the smoothness of \( x \), and \( \lambda \) is a positive value.

In the first term, \( A^{(k)}(d) \in \mathbb{R}^{n \times N} \) is the observation matrix that models the degradation from the high-resolution image \( x \) to the \( k^{th} \) sub-aperture image \( y^{(k)} \), and can be expressed as

\[
A^{(k)}(d) = P_{n \times n}^{(k)} D M^{(k)}(d) B.
\]

Symbol \( P_{n \times n}^{(k)} \) is a matrix that masks pixels in the image according to the structure of \( y^{(k)} \), and is described in detail in Sect. 3.4. Symbol \( D \in \mathbb{R}^{n \times N} \) denotes a pixel sub-sampling operator. As for \( D \), when computing each pixel in the low resolution image, we use bilinear interpolation with the four neighboring pixels in the high resolution image. Symbol \( M^{(k)}(d) \in \mathbb{R}^{N \times N} \) is the translation matrix based on depth map \( d \) for each pixel in the image. These translations consider occlusions, so if two pixels are translated to the same pixel, only the closer pixel is preserved. Symbol \( B \in \mathbb{R}^{N \times N} \) is the PSF for the low resolution images, as in the previous section. Here, color information of \( x \) is handled in the same manner as Eq. (7).

\[
R(x) \text{ in the second term is defined as follows.}
R(x) = \|\nabla_h F x\|_1 + \|\nabla_v F x\|_1,
\]

where \( \nabla_h \) and \( \nabla_v \) are the forward differential operators in the horizontal and vertical directions in the image, respectively. Symbol \( F \) is a matrix converting from RGB space to YUV space. Thus, Eq. (10) is an \( l_1 \) norm for edges, and \( x \) is forced to be sparse with respect to edges.

Equation (8) is minimized using ADMM [27]. Specifically, for \( R(x) \) in the second term of Eq. (8), we define variables \( z_h \in \mathbb{R}^n, z_v \in \mathbb{R}^n \) and corresponding residual variables \( u_h = z_h - \nabla_h F x \), and \( u_v = z_v - \nabla_v F x \), and then repeat updating until these variables converge. With these variables, \( R(x) \) is replaced by the following.

\[
R(x) = \|z_h\|_1 + \|z_v\|_1.
\]

The process to update these variables from \( m^{th} \) to \( (m + 1)^{th} \) is represented as follows.

\[
x^{(m+1)} = \arg \min_x L^x(x^{(m)}, d, z_h^{(m)}, z_v^{(m)}, u_h^{(m)}, u_v^{(m)})
\]

\[
z_h^{(m+1)} = \arg \min_{z_h} L^z_h(z_h^{(m)}, x^{(m+1)}, z_v^{(m)}, u_v^{(m)})
\]

\[
z_v^{(m+1)} = \arg \min_{z_v} L^z_v(z_v^{(m)}, x^{(m+1)}, z_h^{(m)}, u_h^{(m)})
\]

\[
u_h^{(m+1)} = L^u_h(x^{(m+1)}, z_h^{(m+1)}, u_h^{(m)})
\]

\[
u_v^{(m+1)} = L^u_v(x^{(m+1)}, z_v^{(m+1)}, u_v^{(m)}).
\]

Here, \( L^x \), \( L^z_h \), and \( L^z_v \) are defined as follows (* denotes the subscript \( h \) or \( v \)).

\[
L^x(x, d, z_h, z_v, u_h, u_v) = \frac{1}{2} \sum_{k \in \mathcal{K}} \|y^{(k)} - A^{(k)}(d) x\|^2 + \frac{\rho}{2} \|z_h - \nabla_h F x + u_h\|^2 + \frac{\rho}{2} \|z_v - \nabla_v F x + u_v\|^2
\]

\[
L^z_h(z_h, x, u_h) = \lambda \|z_h\|_1 + \frac{\rho}{2} \|z_h - \nabla_h F x + u_h\|^2
\]

\[
L^z_v(z_v, x, u_v) = \lambda \|z_v\|_1 + \frac{\rho}{2} \|z_v - \nabla_v F x + u_v\|^2
\]

\[
L^u_h(z_h, x, u_h) = u_h + z_h - \nabla_h F x
\]

\[
L^u_v(z_v, x, u_v) = u_v + z_v - \nabla_v F x
\]

where \( \rho \) is a positive constant. Equation (17) is differentiable in \( x \), so it can be minimized using a gradient descent method [28]. However, the first term of Eq. (18) is not differentiable in \( z \), so a solution for Eqs. (13) and (14) is computed using a soft thresholding operator \( S \).

\[
z = S_{1/\rho}(\nabla_h F x - u)
\]

\[
S_{\tau}(a) = \begin{cases} a - \tau & (a > \tau) \\ 0 & (|a| \leq \tau) \\ a + \tau & (a < -\tau) \end{cases}
\]

Multiplying \( A^{(k)}(d), \nabla_h, \nabla_v, F, \) and \( S \) from the right can be implemented with image processing operations, so there is no need to keep an \( N \times N \) matrix, which would require large amounts of memory.

3.3 Alternating Estimation of Depth Map and High Resolution Image

We propose a method of alternatingly estimating the depth map \( d \) and the high resolution image \( x \) through reconstructive super resolution image synthesis, by combining the methods in Sects. 3.1 and 3.2. The initial value \( x^{(0)} \) for the high resolution image \( x \) is first generated using an interpolation based method. Specifically, an image corresponding to \( y^{(k)} \) is created with demosaicing from the RAW image, and an up-sampled image is computed using bicubic interpolation. Then, \( d \) and \( x \) are updated by alternating the depth estimation represented by Eq. (22) and super resolution image synthesis represented by Eqs. (23)-(27) until they converge. The \( m^{th} \) update is represented as follows.

\[
d^{(m+1)} = \arg \min_d E_{\text{depth}}(x^{(m)}, d)
\]

\[
x^{(m+1)} = \arg \min_x L(x) \left(x^{(m)}, d^{(m+1)}, z_h^{(m)}, z_v^{(m)}, u_h^{(m)}, u_v^{(m)}\right)
\]

\[
z_h^{(m+1)} = \arg \min_{z_h} L_h^z(z_h^{(m+1)}, x^{(m+1)}, z_v^{(m)}, u_v^{(m)})
\]

\[
z_v^{(m+1)} = \arg \min_{z_v} L_v^z(z_v^{(m+1)}, x^{(m+1)}, z_h^{(m)}, u_h^{(m)})
\]

\[
u_h^{(m+1)} = L_h^u(x^{(m+1)}, z_h^{(m+1)}, u_h^{(m)})
\]

\[
u_v^{(m+1)} = L_v^u(x^{(m+1)}, z_v^{(m+1)}, u_v^{(m)}).
\]

This procedure is equivalent to introducing a step to estimate depth described in Sect. 3.1 before updating high-resolution image \( x \) with ADMM described in Sect. 3.2. An alternative estimation method for a depth map and high-resolution image has been used in earlier work [9], [25], [29], but there has been no earlier method that considers the pixel arrangement of a light-field camera.
3.4 Role of Pixel Arrangement Matrix $P$

We now describe the pixel arrangement matrix $P$, which we have used to handle the structure of RAW images from a Lytro Illum. Ordinary color images have three color values, R, G, and B, for each pixel. However, in sub-aperture images created using the direct method, each pixel has at most one color value (approximately half of the pixels have no color information to maintain the hexagonal lattice structure). To handle this structure, Eqs. (7) and (9) use $P_{k}^{(k)}$ and $P_{m,n}^{(h)}$. These are diagonal matrices, of which diagonal elements take 1 if the corresponding pixel has a color value and 0 if not. These matrices exclude pixels having no information from the energy function, which enables us to use only the original information in the RAW image of the Lytro Illum. In contrast, for sub-aperture images created using an interpolation-based method, all diagonal elements of $P$ are 1, resulting in an identity matrix. Therefore, our framework described in Sect. 3 can apply to both the direct method and to interpolation based methods.

4. Experiments

We conducted experiments capturing four RAW images (called A, B, C, and D) using a Lytro Illum camera. The RAW images had resolution of 7728 × 5368, and the micro-lens array was 542 × 5368 lenses. Accordingly, the resolution of the sub-aperture images created using the direct method was 1084 × 433. For a conventional interpolation-based method, we used Light Field Toolbox v0.3 [16], which is available on-line. In this case, the sub-aperture image resolution was 626 × 433. This software performs resampling from a hexagonal lattice to a square lattice, so the resolution of sub-aperture images is not the same as the number of micro-lenses. Super resolution processing is applied to the sub-aperture image at the center viewpoint, and the resolution after increasing resolution was set to 2450 × 1634. This resolution was selected to match the output resolution of Lytro Desktop [20], which is the dedicated imaging software for the Lytro Illum.

Next we describe details of parameters for super resolution processing. First, the range of values given for depth estimation had to be decided. For A and B, the disparity between adjacent high-resolution sub-aperture images had a range of −2.26 to 2.25 pixels, and this range was quantized into 20 uniform steps. For C and D, the range of −2.26 to 4.74 pixels was quantized into 29 steps. In both cases, this is equivalent to computing the disparity in sub-pixel units of approximately 0.25 pixels in the resolution of the high-resolution image. The window size used for computing the matching cost in Eq. (6) was 9 × 9 pixels. We tried several values for the number of sub-aperture images used for super resolution image synthesis, and the $\lambda$ parameter in Eq. (8) and the values that seemed the best were selected. Figure 6 shows the synthesis results using the proposed method with $\lambda = 1.0$ and varying the number of sub-aperture images used. It shows that increasing the number of sub-aperture images beyond 81 (9 × 9 viewpoints) degrades the resulting image. This could be caused by the distortion and decreased amount of light around the boundaries of each micro-lens. Sub-aperture images consisting of the light rays from near micro-lens boundaries tends to have lower image quality. Figure 7 shows the synthesis results when using 81 sub-aperture images (9 × 9 viewpoints) and varying the value of $\lambda$. The final parameters were 81 sub-aperture images (9 × 9 viewpoints) and $\lambda = 3$ for Eq. (8). The value of $\rho$ in Eqs. (17) and (18) was set to 1.0.

4.1 Convergence in Super Resolution Image Synthesis

As discussed in Sect. 3.3, the proposed method alternately updates the high-resolution image and the depth map. It is mathematically difficult to show that this process converges. If the observation matrix $A(d)$ in Eq. (8) is invariant, the energy function is guaranteed to monotonically decrease by ADMM, but $d$ is updated in each iteration in the proposed method. We plotted how the energy function $E_{SR}$ in Eq. (8) changes with each iteration in Fig. 8. For comparison, we also plotted the case where depth is estimated only for the first iteration and thereafter held constant. In both cases, the energy function decreased with iterations, and the change settled down after dramatic changes in the first few iterations. This suggests that the method is still effective even
for a relatively small number of updates. In the remainder of the experiments, we used 15 iterations. We also show a comparison of the resulting high-resolution images for the two cases with a fixed depth map and an updated depth map in Fig. 9. The case with updating appears to have a better image quality, although the difference is small. In the remainder of the experiments, we updated the depth map.

For the proposed method, processing of approximately 90 minutes was required to generate one high-resolution image in the case of using 9×9 viewpoint input images, updating the depth map, and performing 15 iterations. For these tests, the computing environment was Windows 7 Professional for the OS, an Intel Core i7-4771 3.50 GHz CPU, and 8.0 GB of main memory, and programming was done in C++.

4.2 Evaluating Image Quality

Results using the proposed method for four data sets are shown in Fig. 10. In Fig. 11, close-ups of the super-resolved images (shown with blue squares in Fig. 10) are compared with other super resolution image synthesis methods. We have included only a part of the results in this paper due to limitations of the pages and file size, but the original test result data can be obtained from our Web site[30]. For (i), sub-aperture images were created using an interpolation-based method[16] and up-sampled using bicubic interpolation. For (ii), sub-aperture images were created using an interpolation-based method[16] and super resolution image synthesis was applied. For (iii), sub-aperture images were created using the direct method, and super resolution image synthesis was applied, which is the proposed method. The other two samples included for comparison were created with available software. The first uses the method[19] that generates a high-resolution refocused image based on ray tracing, for which the source code was available. The second was created by Lytro Desktop[20], which is software included with the Lytro Illum. In contrast with other
competitors that increased clarity at all depths, the software in [19] increased clarity at a specified depth. The image in (iv) was adjusted so that the target parts are as clear as possible with visual observation. Lytro Desktop loads the RAW data as input and achieves various functions such as changing the depth of virtual focus, and creating images from different viewpoints. The output result of using Lytro Desktop to sharpen the entire image (all focal-point image) is shown in (v). The image created using the proposed method reproduces small details, with the highest resolution and clarity of all methods except for that produced by Lytro Desktop. Lytro Desktop is only provided as a binary, so it was not possible to check the algorithms used, but it seems to be using some type of image sharpening process. On the other hand, although our proposed method does not perform any post processing to touch-up the appearance, it achieved image quality comparable to Lytro Desktop.

We finally show close-ups of some occlusion boundaries in the resulting images (yellow squares in Fig. 10) in Fig. 12. Generally when processing multi-view images, the handling of occlusion boundaries is important. For the proposed method, depth estimation in Sect. 3.1 performs no particular processing for occlusions. However, the quality in regions with occlusion boundaries is not particularly worse than other regions as shown in Fig. 12. This may be because the light field camera used for this research can provide sufficiently many multi-view images, and the disparity between these images is quite small. As a result, the effects of occlusions would not appear prominently.

5. Extraction of Multi-View Images from Raytrix

We also mention another type of light field cameras, Raytrix, and describe how to extract high-quality multi-view images [31].

The Lytro Illum and Raytrix have a slight difference in terms of the position of the micro-lens array, which significantly affects how to sample the light rays. In the case of a Lytro Illum, each sub-image (consisting of pixels behind each micro-lens) includes only the directional (angular) information. Meanwhile, in the case of Raytrix, each sub-image includes both the spatial and angular information. In fact, each micro-lens can be regarded as a micro camera, as shown in the top of Fig. 13. Moreover, micro-lenses having three different focal lengths are interleaved to constitute the lens array. By exploiting these properties, we can obtain...
Thus, we introduced a sophisticated rendering method for Raytrix cameras in [31]. Specifically, our rendering method is based on the method of Wanner et al. [34] and improved patch size estimation and integration processing of different lens types. We briefly describe our rendering method. We first considered patch size estimation. Wanner et al. applied the Laplacian filters to the patch boundaries to estimate correct patch size but used only the center view image to evaluate the Laplacian. In contrast, we compute the Laplacian for all viewpoint images. This is because the suitable patch size for a specific viewpoint would be suitable also for almost all the viewpoints, and considering all viewpoints together enables us to robustify the estimate of the patch size. We next consider a novel integrating method for microlenses having different focal lengths. Wanner et al. integrated them by taking the average. It reduces rendering artifacts while simultaneously causing blurring of textures because the amount of blurs is different depending on the microlens types. We thus apply weighted averaging integration, of which weights are determined according to the microlens types. Our software is available from our website [36].

Figure 15 shows several rendering results produced by two methods: Wanner et al.’s method [34], the implementation of which is available online [35], and our method [31]. The top row shows the results obtained from a dataset provided by Wanner et al. [34], the second and third rows from datasets provided by Palmieri et al. [37], and the bottom row from a dataset we captured. Although Wanner et al.’s implementation worked well with their dataset as shown in the top row of Fig. 15, our implementation can produce shaper results than theirs. With the datasets by Palmieri et al., the superiority of our method over Wanner et al.’s method is quite obvious (Wanner et al.’s method is designed for a grayscale sensor, and thus, it cannot render color images with its original form). Moreover, when our dataset is used as the input, the software provided by Wanner et al. cannot render adequate multi-view images in spite of the fact that we feed the correct configuration parameters to the software. Therefore, Wanner et al.’s implementation is not robust to different camera configurations, whereas ours can robustly produce good multi-view images.

However, our method described here has a room for improvement in terms of how to handle the physical pixel arrangement in the RAW images. In this implementation, we used pixel interpolation for demosaicing and patch extraction. We expect that more careful consideration on the physical pixel arrangement will lead to a better quality of the resulting multi-view images.

6. Conclusion

In this paper, we have proposed a super resolution image synthesis method that considers the physical pixel arrangement used by a typical light field camera such as Lytro Illum. Specifically, we describe a method that first obtains multi-view images from the RAW image while maintaining
the original color mosaic structure and the hexagonal lattice structure for the micro-lens array. Our method then combines these images to increase the resolution, alternating depth estimation and reconstructive super resolution image synthesis based on the estimated depth. In contrast to previous methods, our method does not perform interpolation on the original data, so it can use the original information in the RAW image as faithfully as possible. Experiments showed that the high-resolution images produced by the proposed method were clearer than those created from the output of Light Field Toolbox [16], and refocused images generated using a ray-tracing process [19]. It also achieved image quality equivalent to the all-in-focus image produced by Lytro Desktop [20]. We also mentioned another type of light field camera, Raytrix, and described how to extract high-quality multi-view images from it.

Fig. 15  Generated sub-aperture images. Left: Input image. Middle: Wanner et al.’s method [34]. Right: Our method [31].
References

[1] E.H. Adelson and J.Y.A. Wang, “Single lens stereo with a plenoptic camera,” IEEE Transactions on pattern analysis and machine intelligence, vol.14, no.2, pp.99–106, 1992.

[2] J. Arai, F. Okano, H. Hoshino, and I. Yuyama, “Gradient-index lens-array method based on real-time integral photography for three-dimensional images,” Applied Optics, vol.37, no.11, pp.2034–2045, 1998.

[3] R. N?g, M. Levoy, M. Br?dif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” Computer Science Technical Report CSTR, vol.2, no.11, pp.1–11, 2005.

[4] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, “Dappled photography: Mask enhanced cameras for het- erodyned light fields and coded aperture refocusing,” ACM Transactions on Graphics, vol.26, no.3, pp.69, 2007.

[5] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H.H. Chen, “Programmable aperture photography: Multiplexed light field acquisition,” ACM Transactions on Graphics, p.55, 2008.

[6] T. Georgiev and A. Lumsdaine, “Focused plenoptic camera and rendering,” Journal of Electronic Imaging, vol.19, no.2, pp.021106–021106, 2010.

[7] “Lytro.” http://lytro.com/, 2017.

[8] “Raytrix.” http://www.raytrix.de/, 2018.

[9] F. Perez Nava and J. Luke, “Simultaneous estimation of super- resolved depth and all-in-focus images from a plenoptic camera,” 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, pp.1–4, 2009.

[10] T.E. Bishop and P. Favaro, “The light field camera: Extended depth of field, aliasing, and superresolution,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.34, no.5, pp.972–986, 2012.

[11] S. Fujita, S. Mikawa, K. Takahashi, M. Panahpourtehrani, and T. Fujii, “Extracting multi-view images from multi-focused plenoptic camera,” to appear in International Workshop on Advanced Image Technology (IWAIT), 2016.

[12] Y. Uchida, K. Takahashi, and T. Fujii, “Super-resolution image synthesis from a light field camera using the hexagonal pixel arrangement and sparsity of edges,” International Workshop on Advanced Image Technology (IWAIT), 2016.

[13] Y. Uchida, K. Ohashi, K. Takahashi, and T. Fujii, “Super resolution image synthesis from a light field camera considering its physical pixel arrangement,” IEICE Transactions on Information and Systems, vol.J99-D, no.9, pp.823–835, 2016.

[14] D. Cho, M. Lee, S. Kim, and Y.-W. Tai, “Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction,” IEEE International Conference on Computer Vision (ICCV), pp.3280–3287, 2013.

[15] K. Ohashi, K. Takahashi, and T. Fujii, “Joint estimation of high resolution images and depth maps from light field cameras,” IS&T/SPIE Electronic Imaging, pp.9011B–9011B, 2014.

[16] Y. Bok, H.-G. Jeon, and I.S. Kweon, “Geometric calibration of micro-lens-based light-field cameras using line features,” European Conference on Computer Vision (ECCV), vol.8694, pp.47–61, 2014.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating di- rection method of multipliers,” Foundations and Trends © in Ma- chine Learning, vol.3, no.1, pp.1–122, 2011.

[18] K. Takahashi and T. Naemura, “Super-resolved free-viewpoint im- age synthesis based on view-dependent depth estimation,” IPSJ Transactions on Computer Vision and Applications (CVA), vol.4, pp.134–148, 2012.

[19] R.C. Hardie, K.J. Barnard, and E.E. Armstrong, “Joint MAP registration and high-resolution image estimation using a sequence of undersampled images,” IEEE Transactions on Image Processing, vol.6, no.12, pp.1621–1633, 1997.

[20] “Lytro.” http://www.fujii.nuee.nagoya-u.ac.jp/~takahashi/Research/Lytro/, 2018.

[21] S. Fujita, S. Mikawa, K. Takahashi, M. Panahpourtehrani, and T. Fujii, “Extracting multi-view images from multi-focused plenoptic camera,” to appear in International Workshop on Advanced Image Technology (IWAIT)/International Forum on Medical Imaging in Asia (IFMIA), 2019.

[22] A. Lumsdaine and T. Georgiev, “The focused plenoptic cam- era,” IEEE International Conference on Computational Photography (ICCP), pp.1–8, 2009.

[23] T.G. Georgiev and A. Lumsdaine, “Reducing plenoptic camera art- ifacts,” Computer Graphics Forum, vol.29, no.6, pp.955–968, 2010.

[24] S. Wanner, J. Fehr, and B. Jahn, “Generating EPI representations of 4D light fields with a single lens focused plenoptic camera,” International Conference on Advances in Visual Computing, vol.6938, pp.90–101, 2011.

[25] “Lytro Desktop.” https://lytro-desktop.foamobile.com/, 2018.

[26] “Lytro Desktop.” https://lytro-desktopsoftware.informer.com/, 2018.

[27] “Lytro Desktop.” https://lytro-desktopsoftware.informer.com/, 2018.

Appendix A: Matrix Representation of Image Processing

In Sect. 3, 2D color image data was represented as 1D vec- tors for the sake of explanation. The elements of the 1D vector can be freely mapped to the pixel positions and color channels in the image. Applying linear image processing to a vectorized image $x$ can be expressed as:

$$ y = Ax, $$

(A.1)
where $y$ is the vector representation of the image after processing. The $j^{th}$ element of $y$ is expressed as:

$$y_j = \sum_i a_{ji} x_i,$$

(A-2)

where $x_i$ represents the $i^{th}$ element of $x$. Symbol $a_{ji}$ represents the element in the $j^{th}$ row and $i^{th}$ column of $A$.

Each element $a_{ji}$ of the matrix $A$ is determined according to the physical meaning of the image processing based on the rule associating the original 2D color image with the 1D vector representation. All of the image processing used in Sect. 3 are represented by the manipulation of the elements $a_{ji}$, including the up-sampling by nearest neighbor interpolation ($U$), the sub-sampling with sub-pixel interpolation ($D$), the translation ($M^k$), the PSF ($B$), the neighborhood area extraction ($W_i$), and the masking to match pixel structures ($P(x, N \times N)$, $P(x, n \times n)$).

First, we consider up-sampling by nearest-neighbor interpolation and translation. As a result of this process, if the $i'$th element $x_{i'}$ of the input vector corresponds to the $j^{th}$ element $y_j$ of the output vector, $a_{ji}$ is expressed as follows:

$$a_{ji} = \begin{cases} 1 & (i = i') \\ 0 & (i \neq i') \end{cases}.$$  

(A-3)

Next, we consider sub-sampling with sub-pixel interpolation and PSF. If the $j^{th}$ element $y_j$ of the output vector is the weighted sum of multiple elements $x_i (i \in I)$ of the input vector, and each weighting is $w_{ji}$, then $a_{ji}$ is expressed as follows:

$$a_{ji} = \begin{cases} w_{ji} & (i \in I) \\ 0 & (i \notin I) \end{cases}.$$  

(A-4)

Finally, we consider masking, where $A$ is a square diagonal matrix. If the set of elements remaining after the process is $I$, then $a_{ii}$ is expressed as follows.

$$a_{ii} = \begin{cases} 1 & (i \in I) \\ 0 & (i \notin I) \end{cases}.$$  

(A-5)

Shu Fujita received a B.E. and M.E. degrees in information engineering in 2014 and 2016 from Nagoya Institute of Technology, Nagoya, Japan, where he is currently working toward a Ph. D graduate degree in electrical engineering and computer science. He is working on light field acquisition and processing.

Keita Takahashi received B.E., M.S., and Ph.D. degrees in information and communication engineering from the University of Tokyo, Tokyo, Japan, in 2001, 2003, and 2006. He was a project assistant professor at the University of Tokyo from 2006 to 2011 and was an assistant professor at the University of Electro-Communications from 2011 to 2013. He is currently an associate professor at the Graduate School of Engineering, Nagoya University, Nagoya, Japan. His research interests include computational photography, image-based rendering, and 3-D displays.

Toshiaki Fujii received B.E., M.E., and Dr.E. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1990, 1992, and 1995. In 1995, he joined the Graduate School of Engineering, Nagoya University, where he is currently a professor. From 2008 to 2010, he was with the Graduate School of Science and Engineering, Tokyo Institute of Technology. His current research interests include multidimensional signal processing, multi-camera systems, multi-view video coding and transmission, free-viewpoint television, and their applications.