

# Spatio-angular consistent editing framework for 4D light field images

Williem<sup>1</sup> · Ki Won Shon<sup>1</sup> · In Kyu Park<sup>1</sup> 

Received: 19 January 2016 / Revised: 30 May 2016 / Accepted: 30 June 2016 /  
Published online: 15 July 2016  
© Springer Science+Business Media New York 2016

**Abstract** This paper presents a practical framework for various light field editing algorithms, such as segmentation, sparse edit propagation, and inpainting. A novel framework is introduced with its user interface to perform the light field editing algorithms. Unlike single-image algorithms, the coherence between light field subaperture images is fully considered. Instead of processing all light field subaperture images independently, the proposed framework performs editing in the cluster image domain. The edit result in the cluster image is propagated back to each light field subaperture image using 2D-to-4D light field edit propagation. Experimental results on test images captured by a Lytro off-the-shelf light field camera confirm that the proposed method provides robust and consistent results of edited light field subaperture images.

**Keywords** 4D, light field · Spatio-angular consistent · Edit propagation · Segmentation · In-painting

## 1 Introduction

Over the last decade, light field processing has become an active research topic that has attracted many researchers in computer vision and computer graphics communities. A light

---

✉ In Kyu Park  
pik@inha.ac.kr

Williem  
22112195@inha.edu

Ki Won Shon  
kiwonshon86@gmail.com

<sup>1</sup> Department of Information and Communication Engineering, Inha University, Incheon, South Korea

field image with 4D spatial and angular parameterizations contains more directional information of captured rays than an ordinary image while sacrificing image resolution [13, 19, 24]. Unlike a conventional image, the focus plane, depth of field, and viewpoint are not fixed at the moment of the image capture but can be adjusted while rendering during post processing. Moreover, the aperture can be synthetically controlled to produce various effects of imaging. In this context, it is expected that a light field camera can be used as a new type of computational camera for capturing the additional information necessary for advance image processing and understanding.

Prior to the availability of consumer light field cameras in the market, it was impractical for a common user to capture a 4D light field because it requires a large-size camera array [28]. However, since Lytro [23] and Raytrix [25] light field cameras were introduced to consumers, light field images have become readily available for numerous applications. As the popularity of the light field continues to grow, the demand for efficient algorithms to process the 4D light field also increases. Various algorithms and applications for consumer light field images, such as depth estimation [5, 27], saliency detection [20], and calibration [4, 8, 11], have been presented in recent academic conferences.

Compared to the maturity of single-image editing algorithms (i.e. matting, segmentation, inpainting, and colorization) [1, 6, 9, 17, 18, 26, 29], light field image editing is still underdeveloped in terms of practicability and user-friendliness. Conventional image-editing algorithms cannot be applied directly to light field images because they do not preserve the angular coherence. In addition, their direct extension causes inefficiency due to the massive redundancy and immense size of the light field image. Although there is a study on light field editing interfaces or tools [14], it is still limited to direct image editing (i.e. pen or brush tools). On the other hand, recent sparse edit propagation on light fields [15] is restricted to synthetic data experiments and requires user input over the 3D representation of the light field with accurate depth information.

In this paper, we propose a novel framework for an efficient editing algorithm, which is performed on a light field image. The editing tool consists of several edit propagation operators (i.e. re-colorization and segmentation) and advance image editing operator (i.e. inpainting). Contrary to [15], our framework simply requires a user input over a 2D image at the center position of a 4D light field, which is a more practical and generalized method for consumers. Note that our tool is designed for various applications of light field edit algorithms. The experiments are conducted using real light field images captured by a Lytro off-the-shelf light field consumer camera.

Our key observation is that, since light field contains high redundancy, the editing can be performed on a representative image (cluster image) to increase the efficiency. While the state-of-the-art methods [3, 15, 29] utilize affinity-based clustering methods or stroke sampling to reduce the complexity, we perform variance-based correspondence matching to select the best cluster for each pixel [27] (all-focus image generation). Then, a state-of-the-art image editing operator is performed on the cluster image. The edit result is propagated back to the 4D light field image by performing 2D-to-4D light field edit propagation. To demonstrate the generality, our study integrates five types of image processing operators: local edit propagation [12], global edit propagation [1], hard segmentation [26], soft segmentation (matting) [18], and inpainting [10]. We have observed that the proposed framework preserves the angular consistency between light field subaperture images due to its clustering method.

In summary, our specific contribution is to propose the followings.

- General framework applicable for various light field image editing operators.

- Novel angular consistency term for 4D light field image editing.
- Practical 2D-to-4D propagation algorithm for a commercial light field camera (Lytro Illum).

The remainder of this paper is organized as follows. Section 2 introduces the related studies. The proposed method is described in Section 3. Section 4 and Section 5 present the applications and experimental results of the proposed framework. Finally, we provide concluding remarks in Section 6.

## 2 Related work

Initially, editing algorithms employed the color similarity between spatial neighboring pixels to locally propagate the sparse user strokes. A sparse affinity matrix was constructed as utilized in colorization [17] and image tonal manipulation [22]. Chen et al. [6] preserved the manifold structure utilizing neighboring pixels in the feature domain. The affinity matrix is solved by the least square solver for a sparse linear system when the number of neighboring pixels is adequate. These methods are computationally expensive for a large image size. An and Pellacini [1] propose a global propagation method by considering all-pair pixel correspondence. However, it is still inefficient to propagate the edit value on an immense light field due to high complexity.

There are two classes of propagation algorithms that speed up the computation. Certain algorithms perform the propagation in the cluster domain [3, 29]. The adaptive k-d tree clustering was employed in the affinity space to optimize [1] with better efficiency [29]. Bie et al. [3] presented a stroke sampling method to reduce the user-stroke sensitivity identified in [29]. Another class of algorithms attempts to replace the global optimization method with a local smoothing function [10, 12, 16, 21]. Li et al. [21] utilized radial basis functions to interpolate the user-edited pixels. A novel domain transform [12], which preserves the edge information, consists of a fast scribble propagation that achieves comparable results with conventional methods. The usage of a domain transform to preserve temporal consistency in graphics applications was performed by Lang et al. [16]. A real-time performance was achieved with the exploitation of GPGPU parallelism. Yet these algorithms are not suitable for light field because they do not consider angular coherence.

Jarabo et al. extended [1] with utilizing the light field based affinity matrix and solving the optimization on feature-based clusters. However, [15] does not consider the coherency in angular domain across the subaperture images and evaluate the performance on synthetic data only. Moreover, it requires for users to provide input strokes in the 3D space, which is impractical for users who want the interaction on 2D images. It also highly depends on the depth information and its accuracy to propagate the 3D strokes into 4D light field image. The work is limited to sparse edit propagation and cannot be extended for advance image edit operator, such as inpainting. [14] performs a user study on how people edit the light field. They introduce various workflows for performing light field editing with two user interfaces: multi-view paradigm and focus paradigm. However, they limit their study on local point-and-click tool instead of studying sparse edit propagation. Ao et al. [2] introduces a novel light field reparametrization to perform downsampling-upsampling for light field editing applications. Nevertheless, their work is limited to the global edit propagation application.

There are numerous soft and hard segmentation algorithms for sparse stroke propagation. Tang et al. [26] applied graph cut optimization to segment an image to several regions.

On the other hand, Levin et al. [18] proposed a closed-form solution based on local color smoothness prior to extracting the alpha map for each region. Both techniques are effective for a single image segmentation but still require more study to deal with light field data. A matting method for light field, which considers angular consistency, was proposed by Cho et al. [7]. However, it requires a trimap as the input instead of user stroke. In addition, the consistent matting with EPI smoothness term is intractable and computationally expensive for computing all light field subaperture images.

Criminisi et al. [9] introduced an efficient inpainting method by filling the missing regions with other patches inside the image. As the inpainting method is highly dependent on the filling order, they propose confidence and data terms to create the best order to inpaint an image. The confidence term leads to find the missing region that has more known pixels and the data term forces to select the missing regions with surrounding linear structures. Their work is well known for a single image inpainting, but it requires an extension to be applied for light field image due to angular consistency. In this paper, we focus on developing a new edit framework for light field that can accommodate all edit operators, while preserving the angular consistency.

### 3 Proposed framework

We develop an efficient editing method for a light field image that preserves angular coherency between subaperture images. A novel light field consistency term  $E_{angular}$  is introduced and integrated with the conventional 2D edit energy function, as described in:

$$E(J) = E_{data}(J) + E_{spatial}(J) + E_{angular}(J) \quad (1)$$

$$J^* = \underset{J}{\operatorname{argmin}} E(J) \quad (2)$$

where  $J$  is the edit value and  $J^*$  is the solution that minimizes  $E(J)$ .  $E_{data}$  and  $E_{spatial}$  are, respectively, the data and spatial smoothness terms which varies depending on the individual edit operator. The new term  $E_{angular}$  enforces the similar results between a pixel and its corresponding pixels in other subaperture images, as defined in:

$$E_{angular}(J) = \sum_{\mathbf{p}} \sum_{\mathbf{q} \in A(\mathbf{p})} (J(\mathbf{p}) - J(\mathbf{q}))^2 \quad (3)$$

where  $\mathbf{p}$  and  $\mathbf{q}$  are the spatial and angular position vectors of a pixel and its corresponding pixel, respectively.  $A(\mathbf{p})$  is the set of the corresponding pixels in the angular domain of a pixel  $\mathbf{p}$ . It determines that a pixel  $\mathbf{p}$  and its corresponding pixel  $\mathbf{q}$  should have the same color since they represent the same point in the real world coordinate. However, the angular (light field consistency) term is intractable because we need to handle a huge sparse affinity matrix. In this paper, we focus on how to approximately solve the light field consistency term efficiently. The overview and detail of the proposed framework are described in the following subsections.

#### 3.1 Framework overview

Figure 1 presents the overview of the proposed framework. The idea behind our method is based on the redundancy nature of light field image. Thus, we observe that it is desirable to perform the image editing algorithms on a cluster image generated from a 4D light field image. Each pixel in a cluster image is expected to have similar value to its corresponding



**Fig. 1** Pipeline of the proposed algorithm (re-colorization example only). Image is captured by Lytro Illum camera

pixels in light field image. We also observe that the all-focus image, generated from the set of refocus images, is suitable to satisfy the demand of the cluster image. This is because each pixel in all-focus image is reconstructed using the corresponding pixels in all subaperture images. In our approach, we use angular color variance to create the all-focus image. The detail of the all-focus image generation is described in Section 3.2. Then, we perform an image editing operator selected by user on the cluster image. In this study, we implement a set of operators including re-colorization (global and local), segmentation (soft and hard), and inpainting.

Next, we need to propagate the edit result on the cluster image to the entire 4D light field. This step is important to achieve consistent results between a pixel and its corresponding pixels as introduced in the angular term. As the cluster image has a pixel value similar to that of light field, we assume that the edit value should be similar too. Note that the assumption is equivalent to having the light field consistency term  $E_{angular}$ . Thus, the idea is to find the corresponding pixel in the cluster image for each pixel in light field subaperture images. We measure the intensity differences to find the corresponding pixels. After the corresponding pixels are found, we utilize the edit value of the corresponding pixel as the edit value of a pixel in the light field image. While the conventional approach [14, 15] relies on the depth accuracy, our method does not require any accurate depth information. We describe the 2D-to-4D propagation method in Section 3.3.

### 3.2 All-focus image generation

We generate the refocus images of the center view to collect the cluster candidates of each pixel. To compute the refocus image  $R_\alpha$  for each cluster candidate  $\alpha$ , each light field subaperture image  $L_O$  is first re-mapped to  $L_\alpha$ :

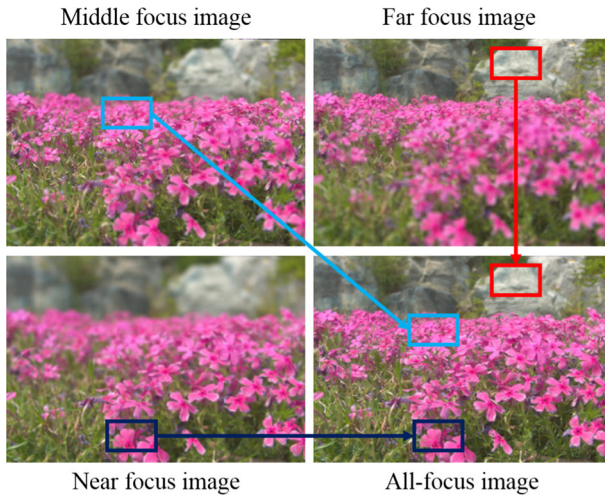
$$L_\alpha(x, y, u, v) = L_O \left( x + u \left( 1 - \frac{1}{\alpha} \right), y + v \left( 1 - \frac{1}{\alpha} \right), u, v \right) \tag{4}$$

where  $(x, y)$  and  $(u, v)$  are the spatial and angular positions for each pixel, respectively. Then, the average of the re-mapped images is computed, as defined in:

$$R_\alpha(x, y) = \frac{1}{W} \sum_{u,v} L_\alpha(x, y, u, v) \tag{5}$$

where  $W$  denotes the number of subaperture images. We compute the angular variance  $\sigma_\alpha^2(x, y)$  of the corresponding pixels for each cluster candidate, as defined in:

$$\sigma_\alpha^2(x, y) = \frac{1}{W} \sum_{u,v} (L_\alpha(x, y, u, v) - R_\alpha(x, y))^2. \tag{6}$$



**Fig. 2** Illustration of all-focus image generation step

Due to the redundancy nature of a light field, the candidate with the minimum variance  $\alpha_{\sigma}^*(x, y)$  is selected as the representative pixel:

$$\alpha_{\sigma}^*(x, y) = \underset{\alpha}{\operatorname{argmin}} \sigma_{\alpha}^2(x, y). \quad (7)$$

Finally, the all-focus image  $I(x, y)$  is described in:

$$I(x, y) = R_{\alpha_{\sigma}^*}(x, y). \quad (8)$$

The clustering method, used in this study, is similar to the variance-based depth estimation method [27]. However, we do not factor in the depth accuracy (cluster index), and hence, it is not required to perform global optimization. Though variance ambiguity might be observed in a textureless region, it does not affect the result because we only need the cluster color intensity instead of the cluster index (depth). Figure 2 shows the illustration of the all-focus image generation step. All-focus image consists of various parts from different refocus images (near focus, middle focus, and far focus images).

### 3.3 2D-to-4D light field edit propagation

Depth invariant light field edit propagation is utilized to transfer the edit value from the cluster image to light field subaperture images. Similar to most image editing techniques, there are two assumptions to propagate the output value from 2D to a 4D light field. The first assumption is that the occluded region has a similar value with neighboring visible regions. The second one is that nearby pixels with similar color should have similar result.

For each pixel in 4D light field, the propagation is performed by computing the most similar pixel in the cluster image. Instead of employing forward propagation, we perform

inverse propagation to ensure that there is no hole in the final result. The candidate pixels are obtained by shifting a pixel in the cluster image for all  $\alpha$ , as described in:

$$I_\alpha(x, y, u, v) = I\left(x - u\left(1 - \frac{1}{\alpha}\right), y - v\left(1 - \frac{1}{\alpha}\right), u, v\right). \tag{9}$$

The absolute intensity difference is computed for each candidate to measure the pixel similarity. We copy the edit value from the most similar pixel. This leads to the following minimization:

$$\alpha_c^*(x, y, u, v) = \underset{\alpha}{\operatorname{argmin}}(I_\alpha(x, y, u, v) - L_O(x, y, u, v))^2 \tag{10}$$

$$J^*(x, y, u, v) = J_{\alpha_c^*}(x, y) \tag{11}$$

where  $J^*$  is the optimum solution. Figure 3 shows the illustration of the propagation step.

### 4 Applications

We show various image editing applications to demonstrate the benefit and general applicability of our framework. The applications are classified into two categories: stroke propagation (edit propagation / segmentation) and inpainting. The workflow is varying for each category and summarized in Fig. 4. The proposed framework can be generalized and it is possible to add new edit operators. Five different algorithms implemented in our framework are described in the following subsections.

#### 4.1 Local and global edit propagation for re-colorization

We employ state-of-the-art algorithms for both local and global edit propagation. In this case,  $J$  in (1) is the color edit value (color difference), denoted as  $s(\mathbf{p})$ . We utilize re-colorization using filtering in the domain transform [12] and An’s colorization algorithm [1] for the local and global edit propagation, respectively. For the local edit propagation, the data and spatial smoothness terms are defined as follows.

$$E_{data}(s) = \sum_{\mathbf{p}} \|s(\mathbf{p}) - t(\mathbf{p})\| \tag{12}$$

$$E_{spatial}(s) = \sum_{\mathbf{p}} \|s(\mathbf{p}) - \sum_{\mathbf{q} \in N(\mathbf{p})} z_{\mathbf{p}\mathbf{q}} s(\mathbf{q})\| \tag{13}$$

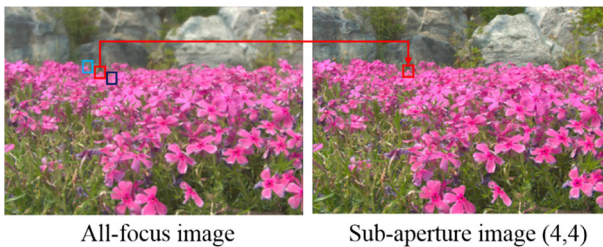
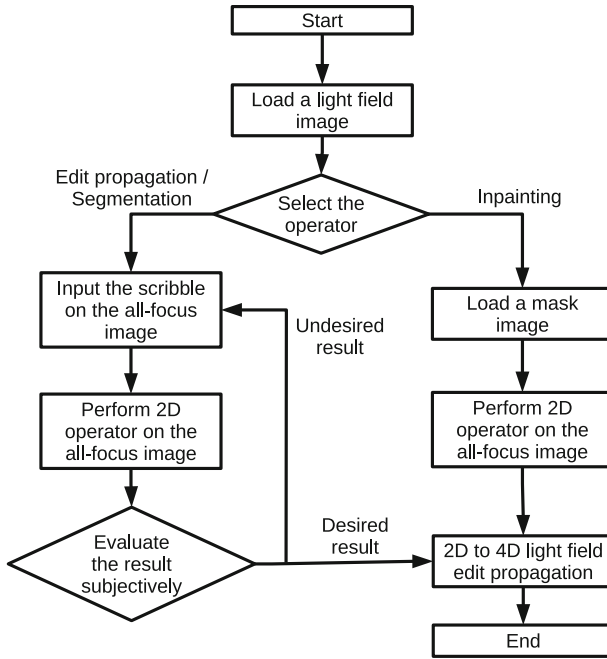


Fig. 3 Illustration of 2D-to-4D light field edit propagation step



**Fig. 4** Workflow of the proposed framework

where  $t(\mathbf{p})$  is the user input and  $N(\mathbf{p})$  is the small neighborhood window centered at  $\mathbf{p}$ .  $z_{\mathbf{p}\mathbf{q}}$  is the affinity value between the pixels  $\mathbf{p}$  and  $\mathbf{q}$ . Furthermore, the data and spatial terms for the global edit propagation are described as follows.

$$E_{data}(s) = \sum_{\mathbf{p}} \sum_{\mathbf{q}} z_{\mathbf{p}\mathbf{q}} w_{\mathbf{p}} (s(\mathbf{p}) - t(\mathbf{p}))^2 \tag{14}$$

$$E_{spatial}(s) = \lambda \sum_{\mathbf{p}} \sum_{\mathbf{q}} z_{\mathbf{p}\mathbf{q}} (s(\mathbf{p}) - s(\mathbf{q}))^2 \tag{15}$$

where  $w_{\mathbf{p}}$  is the weight for the user input  $t(\mathbf{p})$  and  $\lambda = \sum_{\mathbf{p}} w_{\mathbf{p}}/n$  is the relative weight.  $n$  is the number of pixels. Refer to [12] and [1] for the detail. Note that the angular term is same as that in (3). To obtain the final edited light field images  $L_E$ , we add the edit value  $s$  to the original subaperture images  $L_O$ :

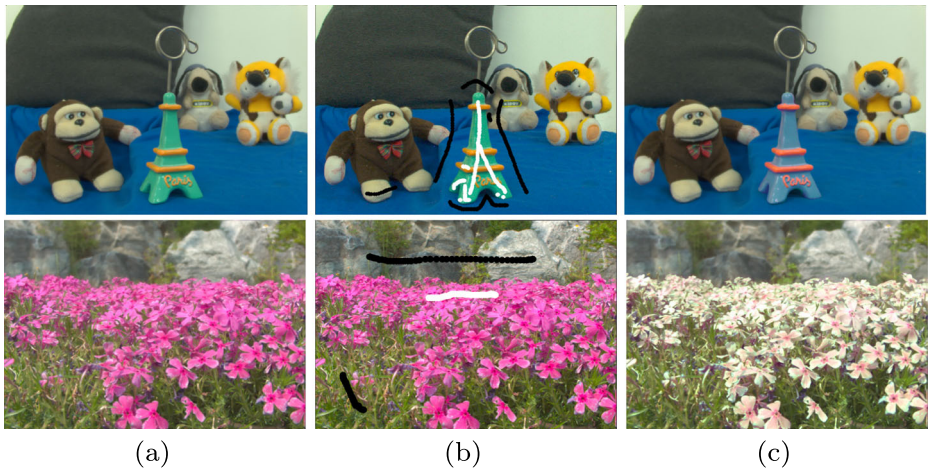
$$L_E(x, y, u, v) = L_O(x, y, u, v) + s(x, y, u, v). \tag{16}$$

Figure 5 shows an example of re-colorization results on the all-focus images.

### 4.2 Hard and soft segmentation

In our framework, we implement both hard and soft segmentation (matting) algorithms. We employ [26] and [18] to perform the hard and soft segmentation, respectively.  $J$  in (1)





**Fig. 5** Results of local (*upper*) and global (*lower*) edit propagation for re-colorization. **a** All-focus images; **b** Input strokes; **c** Re-colored images

is denoted as the alpha value  $a(\mathbf{p})$ . For hard segmentation, we utilize the data and spatial smoothness terms as follows.

$$E_{data}(a) = -\beta \|\theta^S - \theta^{\bar{S}}\| \tag{17}$$

$$E_{spatial}(a) = \lambda \sum_{\mathbf{p}} \sum_{\mathbf{q} \in N(\mathbf{p})} z_{\mathbf{p}\mathbf{q}} |a(\mathbf{p}) - a(\mathbf{q})| \tag{18}$$

subject to hard constraint given by the user input  $t(\mathbf{p})$ .  $\theta^S$  and  $\theta^{\bar{S}}$  are the histograms inside the foreground and background, respectively.  $\beta (= 0.05)$  is the weight of the color separation term and  $\lambda (= 0.95)$  is the weight of the smoothness term.  $z_{\mathbf{p}\mathbf{q}}$  is the affinity value between the pixel  $\mathbf{p}$  and  $\mathbf{q}$ . Furthermore, the data term for soft segmentation is defined as follows.

$$E_{data}(a) = \sum_{\mathbf{p}} \sum_{\mathbf{q} \in N(\mathbf{p})} (a(\mathbf{q}) - I(\mathbf{q})f(\mathbf{p}) - b(\mathbf{p})) \tag{19}$$

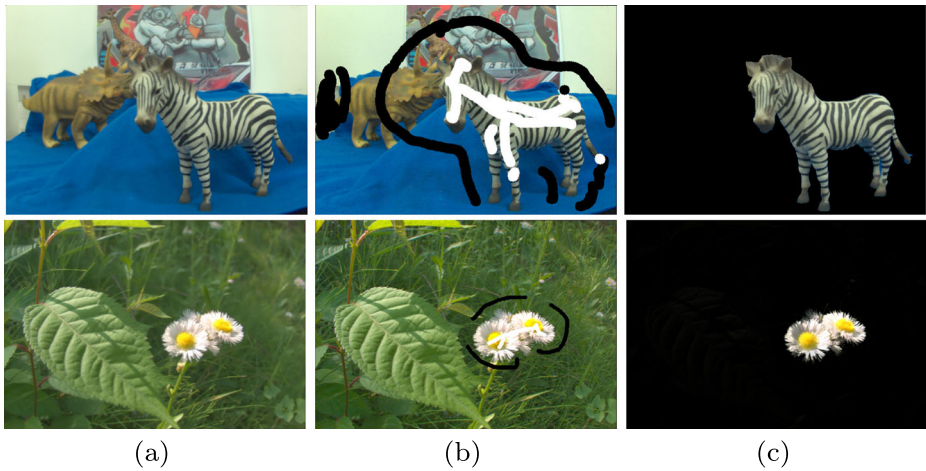
where  $f(\mathbf{p}) = \frac{1}{F(\mathbf{p}) - B(\mathbf{p})}$  and  $b(\mathbf{p}) = -\frac{B(\mathbf{p})}{F(\mathbf{p}) - B(\mathbf{p})}$ .  $F(\mathbf{p})$  and  $B(\mathbf{p})$  are the foreground and background intensities at pixel  $\mathbf{p}$ , respectively.  $I(\mathbf{q})$  is the intensity of the all-focus image at pixel  $\mathbf{q}$ , which is inside a small neighborhood window  $N(\mathbf{p})$ . The spatial smoothness term is as follows.

$$E_{spatial}(a) = \lambda \sum_{\mathbf{p}} f(\mathbf{p})^2 \tag{20}$$

subject to hard constraint given by the user input  $t(\mathbf{p})$  with  $\lambda (= 0.0001)$ . Refer to [26] and [18] for the detail. The segmented 4D light field  $L_S$  is obtained by multiplying the original subaperture images  $L_O$  and the alpha value  $a$ :

$$L_S(x, y, u, v) = L_O(x, y, u, v) \times a(x, y, u, v). \tag{21}$$

Figure 6 shows an example of segmentation results on the all-focus images.



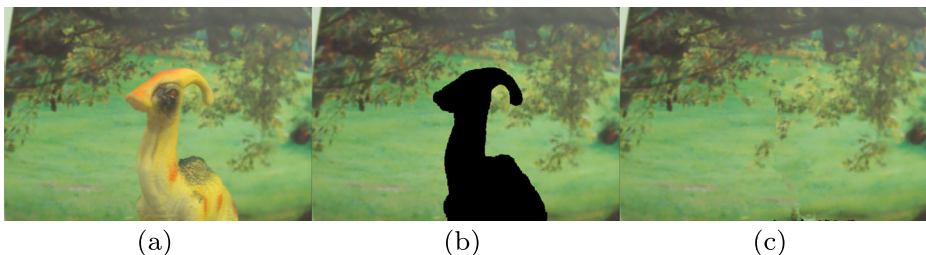
**Fig. 6** Results of hard (*upper*) and soft (*lower*) segmentation. **a** All-focus images; **b** Input strokes; **c** Segmented objects

### 4.3 Inpainting

The proposed framework also accommodates a light field consistent inpainting algorithm. First, an exemplar-based image inpainting algorithm [9] is applied on the cluster image. In this application, we do not simply propagate the result from a 2D cluster image to a 4D light field image. Instead, we perform the inpainting algorithm [9] with search space modification on each light field subaperture image. To preserve the light field consistency and reduce the computational complexity, our inpainting method does not search for the corresponding patch in the whole light field but looks for the best corresponding patch in the inpainted cluster image. The inpainting result is shown in Fig. 7.

## 5 Experimental results

The proposed framework is implemented on an Intel i7 4770 @ 3.4 GHz computer with 12GB RAM. We evaluate the proposed framework with light field data captured by a Lytro Illum camera in indoor and outdoor environments. To extract the 4D light field image, we utilize the toolbox provided by Dansereau et al. [11]. Specifically, the captured light field

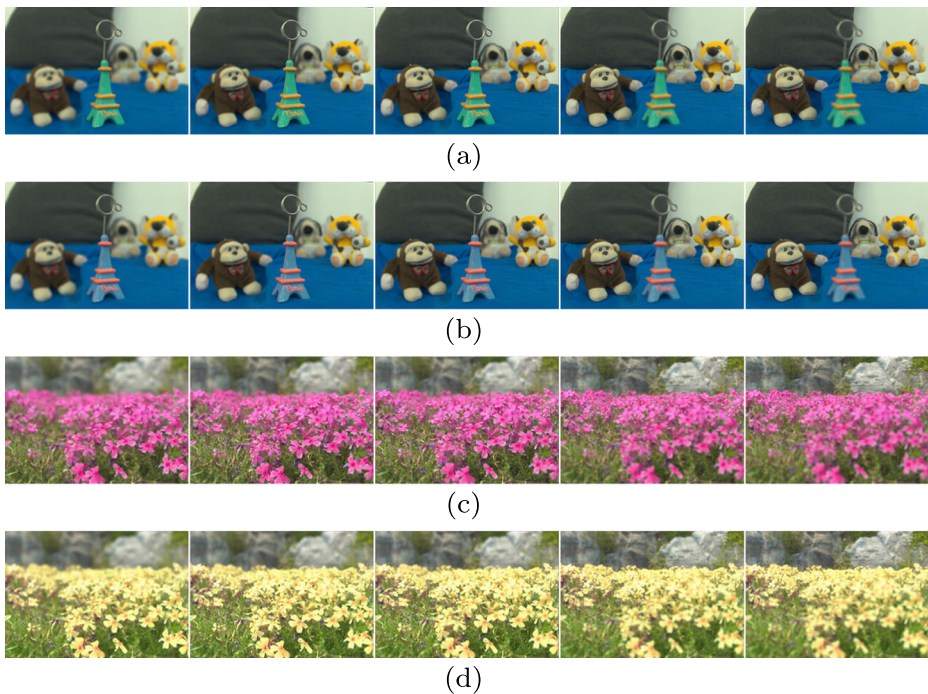


**Fig. 7** Results of inpainting application. **a** All-focus image; **b** Mask image; **c** Inpainted image

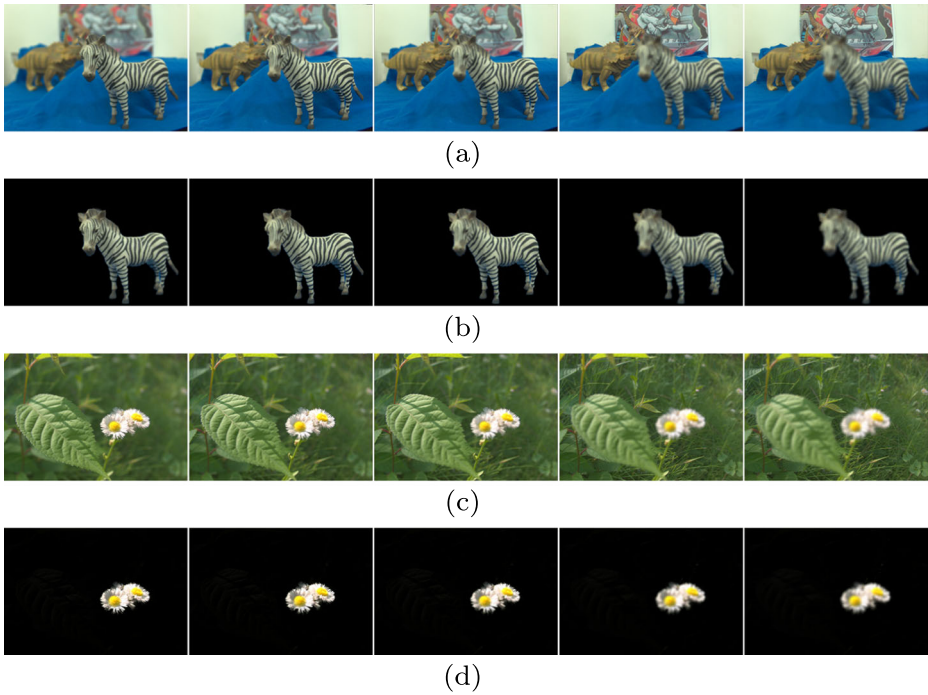
data have a  $625 \times 434$  spatial resolution and a  $15 \times 15$  angular resolution. We utilize 15 different  $\alpha$  values. The input stroke is given on the cluster image. The algorithm is implemented using C++ while a few computationally complex functions are parallelized on the GPU. With the unoptimized implementation, the total running time of 225 subaperture images for a re-colorization operator is approximately 7.1 seconds. These are split into 3.2 seconds for the clustering function, 0.5 seconds for the 2D edit propagation function, and 3.3 seconds for the 4D edit propagation function. Note that the average running time for an individual subaperture image is around 0.032 seconds which is fast enough for an image processing computation time. We evaluate our framework on several applications, as described in Section 4.

### 5.1 Qualitative evaluation

Owing to space limitations, we could not show all light field subaperture images in this paper. Instead, we show the edited refocus images to appraise the edited results. Note that the edited refocus images are generated from the edited 4D light field. Therefore, a consistent edited light field leads to consistent refocus images. Figures 8 and 9 show the refocus images of light field datasets (*Tower*, *Flower*, *Animal*, and *Flower2*) for edit propagation and segmentation, respectively. For the inpainting application, Fig. 10 shows the refocus results of *Dino* dataset. It is observed that the edited refocus images have no artifacts and obtain similar refocus areas as the input images. Figure 11 shows the additional results of



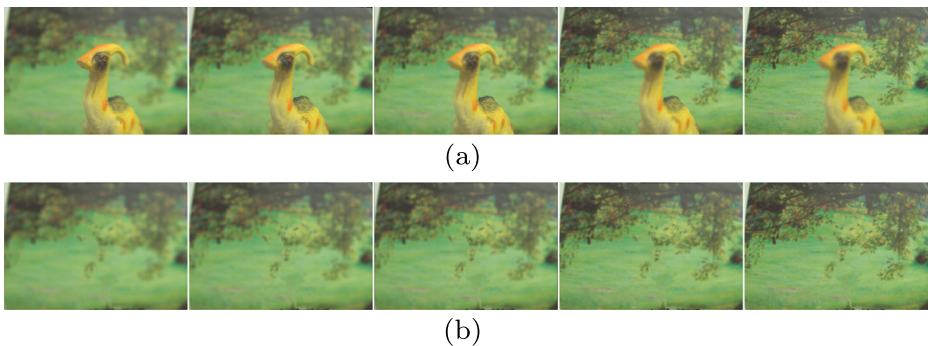
**Fig. 8** Refocus images of original and edited light field. (Left to Right) Near to far focus. **a** *Tower* dataset; **b** Local edit propagation result of (a); **c** *Flower* dataset; **d** Global edit propagation result of (c)



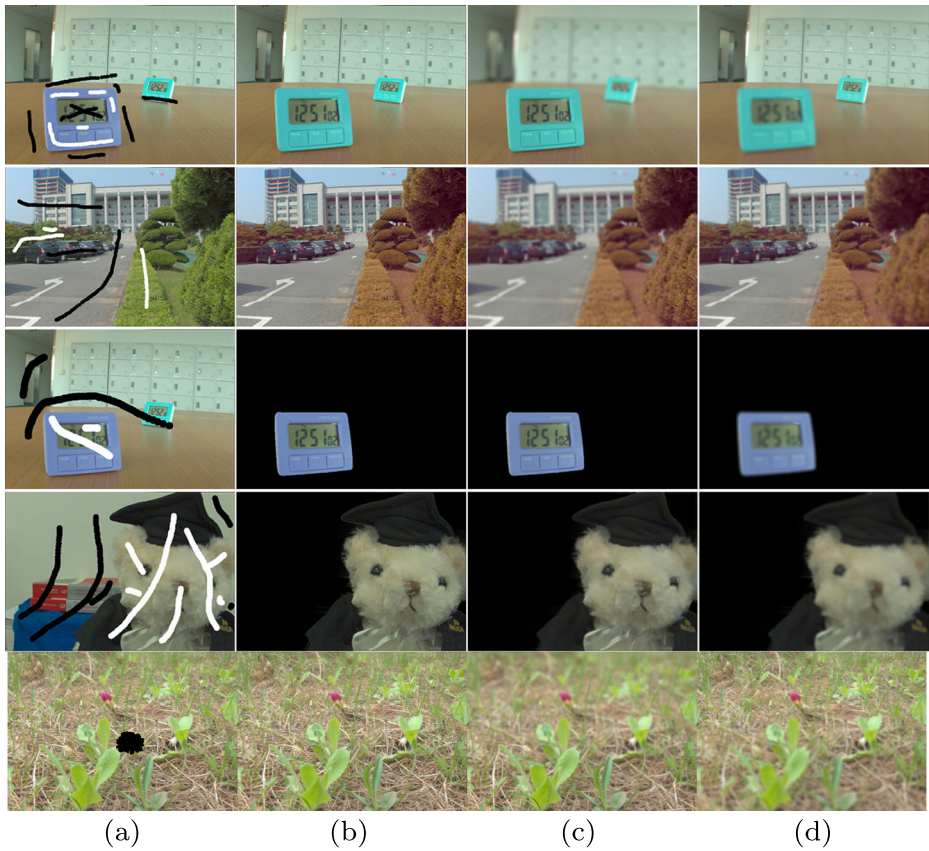
**Fig. 9** Refocus images of original and segmented light field. (Left to Right) Near to far focus. **a** *Animal* dataset; **b** Hard segmentation result of (a); **c** *Flower2* dataset; **d** Soft segmentation result of (c)

other light field datasets. In the supplementary video, we crop some parts of the results and show the zoomed version.

To show the advantages over the existing approach, we evaluate the performance of the single image algorithm without angular consistency. Instead of propagating the edited result from the all-focus image, we first propagate the user input to each light field sub-aperture image. Then, we apply the single image algorithm for each sub-aperture image. Figures 12

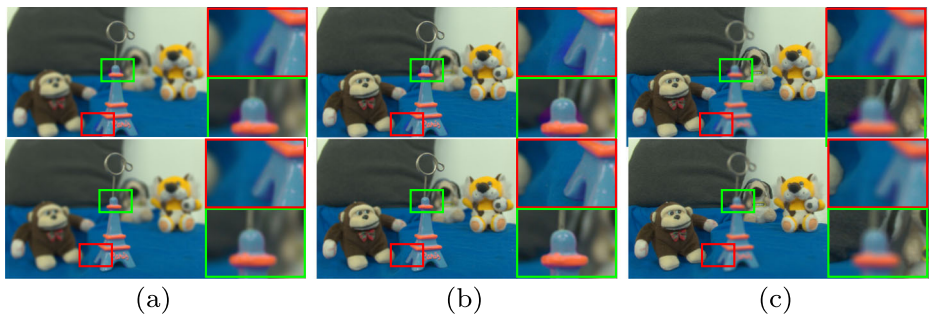


**Fig. 10** Refocus images of the original and inpainted light field. (Left to Right) Near to far focus. **a** *Dino* dataset; **b** Inpainting result of (a)

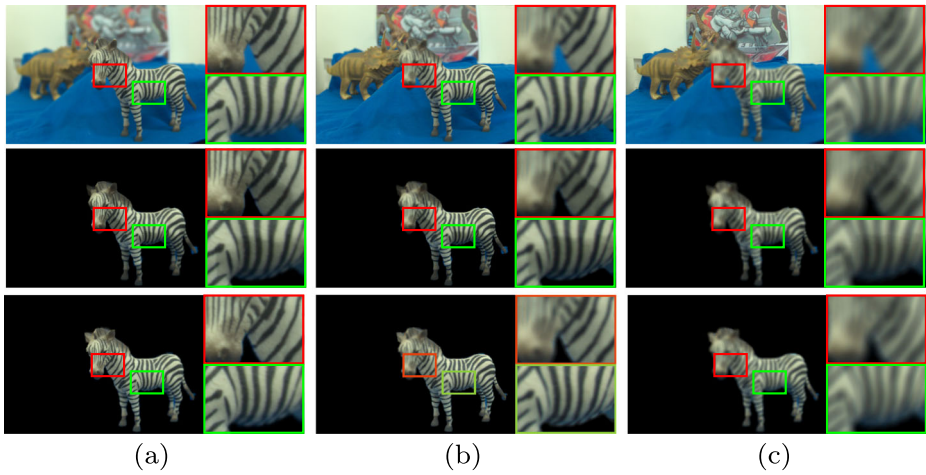


**Fig. 11** Additional results of our editing framework. **a** Cluster image with user stroke; **b** Edited cluster image; **c** Edited near focus image; **d** Edited far focus image; (First row) Local edit propagation; (Second row) Global edit propagation; (Third row) Hard segmentation; (Fourth row) Soft segmentation; (Fifth row) Inpainting

and 13 show the comparison of the local edit propagation for recolorization and hard segmentation results, respectively. We show that the existing approaches have some artifacts



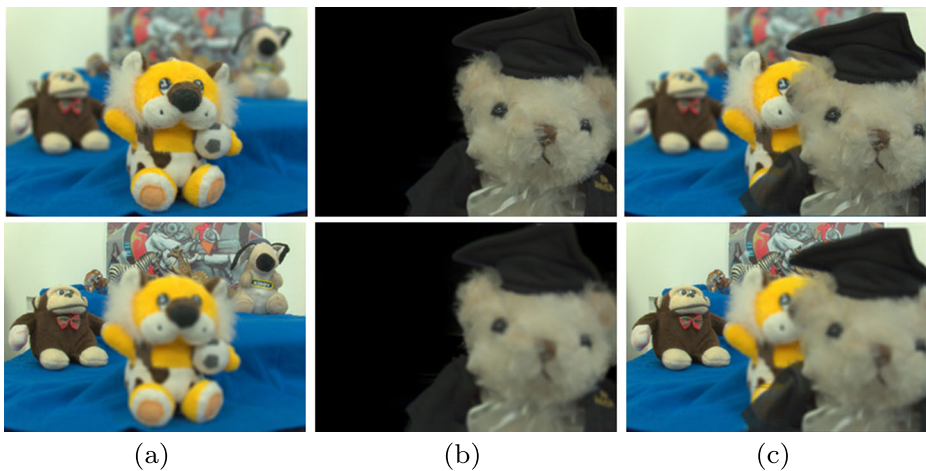
**Fig. 12** Comparison of local edit propagation application. **a** Near focus; **b** Middle focus; **c** Far focus; (Upper) Conventional framework; (Lower) Proposed framework



**Fig. 13** Comparison of hard segmentation application. **a** Near focus; **b** Middle focus; **c** Far focus; (*Upper*) Input images; (*Middle*) Conventional framework; (*Lower*) Proposed framework

because they do not consider angular consistency. For example, we can notice the color leakage around the tower edge boundary in Fig. 12 and the dark artifacts of the zebra body in Fig. 13. The zoomed version of the patches with artifacts is shown for clearer observation.

To show the feasibility of another advanced image processing algorithm on a light field, we develop a prototype of cut and paste algorithm. It integrates two light field images which act as background and foreground, consecutively. An alpha map obtained from soft segmentation operator is required to naturally blend both 4D light fields. Figure 14 presents the example results of cut and paste prototype application. It is apparent that the application generates appreciable refocus images of cut-and-paste results.



**Fig. 14** Example of cut and paste application. **a** Background images; **b** Foreground images; **c** Cut and paste results; (*Upper*) Near focus images; (*Lower*) Far focus images

## 5.2 Computational complexity analysis

We analyze the computational complexity of the global edit propagation for recolorization. For a single image recolorization, the cost of An's colorization algorithm [1] is  $O(m^2n)$  where  $n$  is the number of pixels and  $m$  is the number of samples. Furthermore, as shown in [15], the computational complexity of the colorization algorithm for a light field image is  $O(m^2ln)$ , where  $l$  is the number of sub-aperture images. Although the complexity can be reduced by downsampling, it linearly increases with the light field size.

On the other hand, in the proposed method, the costs for all-focus image generation and 2D-to-4D light field edit propagation are  $O(cln)$  where  $c$  is the number of cluster candidates. Then, the colorization on the all-focus image needs computation with  $O(m^2n)$ . Since  $cl < m^2$ , the total cost of the proposed algorithm is  $O(m^2n)$ , which is smaller than  $O(m^2ln)$ . Therefore, it is shown that the proposed framework is more efficient than conventional approaches.

## 6 Conclusion

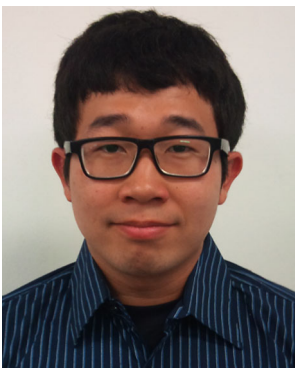
In this paper, we proposed a general framework for image editing algorithms on a 4D light field. To maintain the coherence between light field subaperture images, we presented a novel light field consistency term that was integrated with general edit optimization. Instead of directly solving the light field consistency term, we proposed the editing method in the efficient representative image of immense light field data. We generated the cluster image by utilizing a set of refocus images. Then, various image editing algorithms were performed on the cluster image. After performing the editing process, we propagated the edit information from the cluster image to 4D light field image. The experimental results showed that the proposed method achieved satisfactory results for all edit operators and fast computational time (0.032 seconds for a subaperture image). We believe that our findings on angular-consistency cost approximation will be advantageous the other light field problems as well.

**Acknowledgments** This work was supported by Inha University Research Grant.

## References

1. An X, Pellacini F (2008) Approp: all-pairs appearance-space edit propagation. *ACM Trans Graph* 27(3):40:1–40:9
2. Ao H, Zhang Y, Jarabo A, Masia B, Liu Y, Gutierrez D, Dai Q (2015) Light field editing based on reparameterization. In: *Proceedings of the 16th pacific-rim conference on multimedia*, pp 601–610
3. Bie X, Huang H, Wang W (2011) Real time edit propagation by efficient sampling. *Comput Graphics Forum* 30(7):2041–2048
4. Bok Y, Jeon HG, Kweon IS (2014) Geometric calibration of micro-lens-based light-field cameras using line features. In: *Proceedings of the european conference on computer vision (ECCV)*, pp 47–61
5. Chen C, Lin H, Yu Z, Kang SB, Yu J (2014) Light field stereo matching using bilateral statistics of surface cameras. In: *Proceedings of IEEE computer vision and pattern recognition*, pp 1518–1525
6. Chen X, Zou D, Zhao Q, Tan P (2012) Manifold preserving edit propagation. *ACM Trans Graph* 31(6):132:1–132:7
7. Cho D, Kim S, Tai YW (2014) Consistent matting for light field images. In: *Proceedings of the european conference on computer vision (ECCV)*, pp 90–104
8. Cho D, Lee M, Kim S, Tai YW (2013) Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction. In: *Proceedings of IEEE international conference on computer vision*, pp 3280–3287

9. Criminisi A, Perez P, Toyama K (2004) Region filling and object removal by exemplar-based inpainting. *IEEE Trans Image Process* 13(9):1200–1212
10. Criminisi A, Sharp T, Rother C, P'erez P (2010) Geodesic image and video editing. *ACM Trans Graph* 29(5):134:1–134:15
11. Dansereau DG, Pizarro O, Williams SB (2013) Decoding, calibration and rectification for lenselet-based plenoptic cameras. In: *Proceedings of IEEE computer vision and pattern recognition*, pp 1027–1034
12. Gastal ESL, Oliveira MM (2011) Domain transform for edge-aware image and video processing. *ACM Trans Graph* 30(4):69:1–69:11
13. Gortler SJ, Grzeszczuk R, Szeliski R, Cohen MF (1996) The lumigraph. In: *Proceedings of SIGGRAPH*, pp 43–54
14. Jarabo A, Masia B, Bousseau A, Pellacini F, Gutierrez D (2014) How do people edit light fields? *ACM Trans Graph* 33(4)
15. Jarabo A, Masia B, Gutierrez D (2011) Efficient propagation of light field edits. In: *Proceedings of the fifth ibero-american symposium in computer graphics*, pp 75–80
16. Lang M, Wang O, Aydin T, Smolic A, Gross M (2012) Practical temporal consistency for image-based graphics applications. *ACM Trans Graph* 31(4):34:1–34:8
17. Levin A, Lischinski D, Weiss Y (2004) Colorization using optimization. *ACM Trans Graph* 23(3):689–694
18. Levin A, Lischinski D, Weiss Y (2008) A closed-form solution to natural image matting. *IEEE Trans Pattern Anal Mach Intell* 30(2):228–242
19. Levoy M, Hanrahan P (1996) Light field rendering. In: *Proceedings of SIGGRAPH*, pp 31–42
20. Li N, Ye J, Ji Y, Ling H, Yu J (2014) Saliency detection on light field. In: *Proceedings of IEEE computer vision and pattern recognition*, pp 2806–2813
21. Li Y, Ju T, Hu SM (2010) Instant propagation of sparse edits on images and videos. *Comput Graphics Forum* 29(7):2049–2054
22. Lischinski D, Farbman Z, Uyttendaele M, Szeliski R (2006) Interactive local adjustment of tonal values. *ACM Trans Graph* 25(3):646–653
23. Lytro (2014) The Lytro camera. <https://www.lytro.com>
24. Ng R, Levoy M, Bredif M, Duval G, Horowitz M, Hanrahan P (2005) Light field photography with a hand-held plenoptic camera. In: *Stanford tech report CTSR 2005-02*
25. Raytrix (2013) 3D light field camera technology. [www.raytrix.de](http://www.raytrix.de)
26. Tang M, Gorelick L, Veksler O, Boykov Y (2013) Grabcut in one cut. In: *Proceedings of IEEE international conference on computer vision*, pp 1769–1776
27. Tao M, Hadap S, Malik J, Ramamoorthi R (2013) Depth from combining defocus and correspondence using light-field cameras. In: *Proceedings of IEEE international conference on computer vision*, pp 673–680
28. Wilburn B, Joshi N, Vaish V, Talvala E, Artunze E, Barth A, Adams A, Horowitz M, Levoy M (2005) High performance imaging using large camera arrays. *ACM Trans Graph* 24(3):765–779
29. Xu K, Li Y, Ju T, Hu SM, Liu TQ (2009) Efficient affinity-based edit propagation using K-D tree. *ACM Trans Graph* 28(5):118:1–118:6

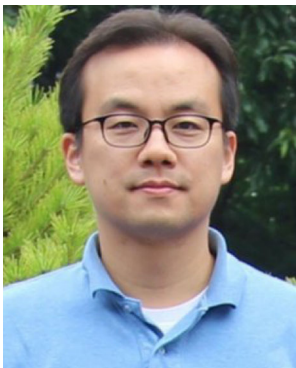


**Williem** received the B.S. degree in computer science from Bina Nusantara University, Indonesia in 2011. He is currently working toward Ph.D. degree in Inha University, Korea. His research interests include computer vision, computational photography, and GPU computing.





**Ki Won Shon** received the B.S. and M.S. degrees in information and communication engineering in Inha University, in 2013 and 2015, respectively. Currently he is a researcher in Zetta Imaging.



**In Kyu Park** received the B.S., M.S., and Ph.D. degrees from Seoul National University (SNU) in 1995, 1997, and 2001, respectively, all in electrical engineering and computer science. From September 2001 to March 2004, he was a Member of Technical Staff at Samsung Advanced Institute of Technology (SAIT). Since March 2004, he has been with the School of Information and Communication Engineering, Inha University, where he is a full professor. From January 2007 to February 2008, he was an exchange scholar at Mitsubishi Electric Research Laboratories (MERL). From September 2014 to August 2015, he was a visiting associate professor at MIT Media Lab. Dr. Park's research interests include the joint area of computer graphics and vision, including 3D shape reconstruction from multiple views, image-based rendering, computational photography, and GPGPU for image processing and computer vision. He is a senior member of IEEE and a member of ACM.