LIGHT FIELD IMAGE EDITING BY 4D PATCH SYNTHESIS

Ke-Wei Chen   Ming-Hsu Chang   Yung-Yu Chuang

National Taiwan University

ABSTRACT

This paper presents a patch-based synthesis framework for lightfield image editing. The core of the proposed method builds upon a patch-based optimization approach. The main contribution of the paper is to extend the versatile patch-based image editing framework to 4D lightfield images and enable many editing applications for them. Specifically, the paper introduces a novel 4D lightfield patch consistency measure for avoiding synthesis of inconsistent patches into the edited lightfield images. Combining with a joint 4D patch search, our method is able to maintain the correlation among views and render a consistent interpretation of the scene. The proposed method offers patch-based solutions to a wide variety of lightfield image editing problems, including inpainting, retargeting and reshuffling.

Index Terms— Lightfield images, image editing, patch synthesis.

1. INTRODUCTION

Lightfield cameras have gained a lot of attention recently because several commercial ones hit the market such as Lytro and Pelican. They capture ligthfield images with different principles including camera arrays (e.g., Pelican) and microlens arrays (e.g., Lytro). A lightfield image records a portion of the 4D lightfield function of the captured scene. Thus, it reveals more information than traditional 2D images. With the richer recorded information, several novel applications beyond what traditional cameras can provide are enabled, such as image refocusing and view interpolation.

As lightfield cameras are becoming popular, manipulating lightfield images could become an important demand. Although lightfield images can essentially be taken as multiple images captured at different views, these images have tightly coupled relationships and independent editing of views can not maintain such relationships. Therefore, editing lightfield images poses two main challenges than editing conventional images. Firstly, the editing results should maintain a consistent scene interpretation, which requires that the corresponding points in different views are adjusted jointly. Secondly, objects with different depths should be handled separately so that they do not affect each other, implying the needs for depth-aware processing.

Since lightfield images are just emerging, there are only few exiting methods for lightfield image editing. All of these methods focus on solving a single editing problem such as inpainting and labeling. This paper extends the versatile patch-based image editing framework from 2D images to 4D lightfield images. The patch-based methods have been proved useful in many editing problems, such as retargeting, inpainting and rearrangement. They have been even implemented in Adobe Photoshop. By extending the versatile editing framework to lightfield images, our method enables a rich set of editing operations to the domain of lightfield images.

2. RELATED WORK

Image editing. Patch-based methods have been widely applied to several image editing problems. Criminisi et al. proposed a patch-based method for image inpainting [1] and it can be taken as one of the early example for patch-based image editing. Simakov et al. proposed the bidirectional similarity [2] to measure the distance between two images and used it in several applications including summarizing and retargeting images. Our method is built upon this measure but extends it to 4D domain. Pritch et al. proposed the shift-map method [3]. The method finds an optimal shift for each pixel so that the shift-map is a rearrangement of the input image for the target editing. Each shift vector is mapped to a label; based on these labels, the image editing problem is transformed into a graph labeling problem. PatchMatch [4] is an efficient randomized algorithm to accelerate the process of finding the most similar patch. The algorithm first randomly initializes the nearest neighbor field, and then propagates good shifts to adjacent pixels. Good matches can also be generated by randomly choosing candidates with shifts at multiple scales. It has been used in several applications such as the recent image melding [5]. Our approach also uses PatchMatch for optimization, but extends it to 4D patches and augments a new distance function taking depth into account.

Stereoscopic image editing. Stereoscopic images can be taken as the 3D counterpart of the traditional 2D images. Their editing also receives certain amount of attention recently. Stereoscopic image inpainting techniques have been proposed by Wang et al. [6] and Morse et al. [7]. Warping-

This work was partly supported by grants MOST103-2218-E-002-030, NSC101-2628-E-002-031-MY3 and NTU104R7609-5.

based methods have been used frequently for stereoscopic image editing, such as disparity remapping [8] and retargeting [9, 10]. On the other hand, patch-based methods have not been used until very recently when Luo et al. extended PatchMatch to stereoscopic image editing [11].

Lightfield image editing. Lightfield images have received a lot of attention recently because of commercialization of several lightfield cameras such as Lytro and Pelican. Most of these cameras emphasize the application of refocusing. Nevertheless, lightfield images have more applications such as image segmentation and view interpolation. Wanner et al. proposed a method for depth estimation from lightfield images using a variational framework and the epipolar image plane [12]. They also applied the variational framework to multi-label segmentation of 4D lightfield images [13]. Kim et al. proposed a method for reconstructing depth maps of a complex scene from a 3D lightfield [14]. They utilized coherence in the 3D lightfield to estimate depth maps. Some applied the refocusing property in lightfields to improve saliency detection [15]. They made use of the focal stack from lightfields to get a better saliency map. In addition to applications using lightfield images, their editing has also been noticed although only recently. Birklbauer et al. performed lightfield image retargeting based on the seam carving approach [16]. Our method extends the versatile patch-based editing framework to the domain of lightfield images and enables several interesting editing operations for them.

3. 4D PATCH SYNTHESIS FOR LIGHT FIELDS

Depending on the camera used for capturing lightfield images, a lightfield image can be represented with several different forms. A rendering process is often required for obtaining the images for a specific application, such as refocusing. For dealing with different types of lightfield images, the first step of our algorithm is to render a multiview image with $n \times m$ all-in-focus images from the raw lightfield image data. Figure 1 shows an example of a multiview image. We denote a 4D lightfield image as $\mathbf{L}(u, v, x, y)$, where $1 \leq u \leq n$, $1 \leq v \leq m$, $1 \leq x \leq w$, $1 \leq y \leq h$, u and v are view indices, x and y specify a spatial location, $n \times m$ is the angular resolution and $x \times y$ is the spatial resolution. With the multiview representation, the lightfield image can also be represented by a set of images, $\{\mathbf{I}_{uv} | 1 \leq u \leq n, 1 \leq v \leq m\}$ in which $\mathbf{I}_{uv}(x, y) = \mathbf{L}(u, v, x, y)$.

3.1. Depth maps

In addition to color data, the lightfield data also contain depth data implicitly. Such depth data often provide useful auxiliary information for image editing. Thus, we estimate the depth map $\mathbf{D}_{uv}$ for each view (u, v) from the input lightfield $\mathbf{L}$. We estimate the depth using an approach based on the epipolar image plane (EPI) [14]. The lightfield data is 4D and there are several ways to take 2D slices from them. For example, the multiview image $\mathbf{I}_{uv}(x, y)$ is formed through taking the $x-y$ slice by fixing the view index (u, v). One can obtain the horizontal EPI $\mathbf{E}_{vy}(u, x)$ by fixing v and y and taking the $u-x$ slice of the 4D lightfield data. Similarly, the vertical EPI $\mathbf{E}_{ux}(v, y)$ is obtained by fixing u and x and taking the $v-y$ slice. The EPI images better reveal the depth structure of the scene; the corresponding pixels in different views jointly form a line of the same color in the EPI image and the depth is related to the slope of the line.

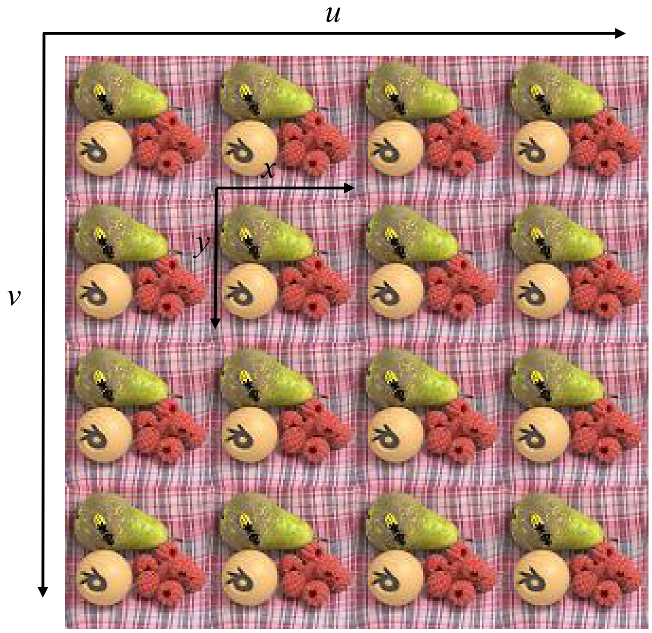


Fig. 1: The multiview representation of a lightfield image. (u, v) is the view index and (x, y) denotes the spatial location within a view.

For a given lightfield index $\mathbf{p} = (\tilde{u}, \tilde{v}, \tilde{x}, \tilde{y})$ and its radiance $\tilde{r} = \mathbf{L}(\tilde{u}, \tilde{v}, \tilde{x}, \tilde{y})$, we estimate its depth using the following procedure. For a hypothesized depth d, we obtain a set of radiance samples $\Gamma(\mathbf{p}, d)$ from the horizontal and vertical EPI images as

$$\begin{aligned} \Gamma(\mathbf{p}, d) = & \{\mathbf{E}_{\tilde{v}\tilde{y}}(\tilde{u}, \tilde{x} + (u - \tilde{u})d) | u = 1..n\} \\ & \cup \{\mathbf{E}_{\tilde{u}\tilde{x}}(\tilde{v}, \tilde{y} + (v - \tilde{v})d) | v = 1..m\}, \end{aligned} \tag{1}$$

in which we find the corresponding pixels in different views by tracing the line formed by the hypothesized depth d. With the color constancy assumption, if the depth estimate d is correct, the sampled radiance values should be consistent to $\tilde{r}$. Thus, we measure the goodness of a depth estimate d by

$$\Delta(\mathbf{p}, d) = \frac{1}{|\Gamma(\mathbf{p}, d)|} \sum_{\gamma \in \Gamma(\mathbf{p}, d)} \psi(\|\gamma - \tilde{r}\|), \tag{2}$$

where the kernel function ψ remaps the difference between

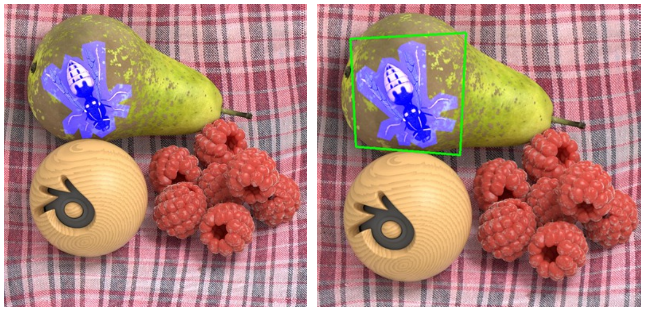


Fig. 2: Region transfer. In (a), the user specifies a region to be removed (the blue region) in one view. The region is automatically transferred to other views. (b) shows the transferred region for another view. The green quadrangle indicates the transferred bounding box.

two colors and we used the same function as Kim et al. [14]:

$$\psi(x) = \begin{cases} 1 - (x/h)^2 & \text{if } (x/h)^2 < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

in which we set $h = 0.02$. The depth for the given index $\mathbf{p}$ is chosen as the one with the best score as

$$\mathbf{D}(\mathbf{p}) = \arg\min_{d} \Delta(\mathbf{p}, d). \tag{4}$$

We do not incorporate any regularization term in depth estimation because we have many views for the data term and the depth obtained this way is already sufficient to our application. Note that the estimated depth is propagated to all corresponding pixels in other views rather than estimated independently. Figure 3(a) shows the estimated depth maps for a couple of views for the lightfield image shown in Figure 1.

3.2. Bidirectional similarity

Similar to other patch-based image editing algorithms, our lightfield image editing framework is based on the bidirectional similarity proposed by Simakov et al. [2], but we extend it to 4D patches from 2D ones. Given two images $\mathbf{S}$ and $\mathbf{T}$, the bidirectional similarity measures their similarity by the following equation:

$$D(\mathbf{S}, \mathbf{T}) = D_{completeness}(\mathbf{S}, \mathbf{T}) + D_{coherence}(\mathbf{S}, \mathbf{T}). \tag{5}$$

The first term $D_{completeness}$ requires that patches in the source image $\mathbf{S}$ are preserved in the target image $\mathbf{T}$ as much as possible so that $\mathbf{T}$ faithfully represents $\mathbf{S}$. It can be defined as the following:

$$D_{completeness}(\mathbf{S}, \mathbf{T}) = \frac{1}{N(\mathbf{S})} \sum_{P \subset \mathbf{S}} \min_{Q \subset \mathbf{T}} \Psi(P, Q), \tag{6}$$

where $N(\mathbf{S})$ is the number of patches in the image $\mathbf{S}$ and $\Psi(P, Q)$ measures the distance between two patches P and Q. For 2D images, the distance is usually defined as the sum of square distances of the intensity values. The second term encourages each patch in the target image $\mathbf{T}$ has a similar counterpart in the source image $\mathbf{S}$ so that the visual coherence is retained and the target image has a similar appearance as the source. It is defined as

$$D_{coherence}(\mathbf{S}, \mathbf{T}) = \frac{1}{N(\mathbf{T})} \sum_{Q \subset \mathbf{T}} \min_{P \subset \mathbf{S}} \Psi(Q, P), \tag{7}$$

The bidirectional similarity can be used in several applications. For image summarization, given the source image $\mathbf{S}$, one could simply find $\mathbf{T}$ which minimizes the metric, i.e., $\mathbf{T} = \arg\min_{\mathbf{T}} D(\mathbf{S}, \mathbf{T})$. The optimization problem can be solved by PatchMatch efficiently [4]. For other editing applications, various constraints could be added into the optimization to meet the requirements of the target application.

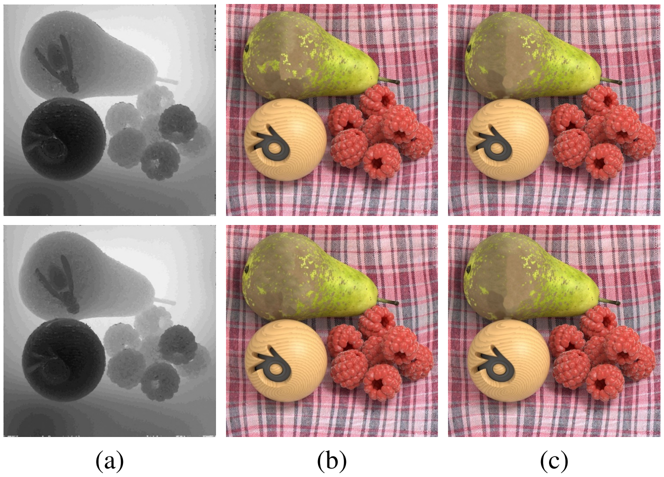


Fig. 3: An example of lightfield image completion. The top row is the view $(1, 1)$ and the bottom row is the view $(1, 5)$. In (a), we show the estimated depth maps. (b) shows the synthesized results if each view is synthesized independently. Notice that the inpainted regions on the pear at different views are not consistent. (c) demonstrates the results of our method which synthesizes views jointly.

3.3. 4D patch synthesis for lightfield

When handling lightfield images, the source $\mathbf{S}$ and the target $\mathbf{T}$ in Equation (5) are 4D hyper-volumes rather than 2D rectangles. In addition, the PatchMatch algorithm needs to be extended to 4D patches. In our current implementation, we use $3 \times 3 \times 7 \times 7$ as the size of 4D patches. Overall, we have made the following major modifications so that the idea of patch synthesis can be applied to lightfield image editing.

Distance function. In addition to color similarity, we also want to maintain similarity of the depth structure. Thus, we measure the distance between two patches based on both their color and depth values. The distance function Ψ is then defined as

$$\begin{aligned}\Psi(P,Q) = &\sum_{\mathbf{p},\mathbf{q}} \|\mathbf{L}(\mathbf{p}) - \mathbf{L}(\mathbf{q})\|^2 \\ &+ \lambda \sum_{\mathbf{p},\mathbf{q}} |\mathbf{D}(\mathbf{p}) - \mathbf{D}(\mathbf{q})|^2,\end{aligned} \tag{8}$$

where $\mathbf{p}$ and $\mathbf{q}$ are a pair of corresponding positions in 4D patch P and patch Q; λ balances between the importance of both color and depth consistency and we empirically set $\lambda = 0.2$ in our experiments. With the new distance, our method also pays attention to depth structure when searching for the best matching patches, making the method more depth-aware.

Patch consistency. In addition to the completeness and coherence terms, we add a patch consistency into the energy function in Equation (5). This term is designed to measure whether a patch is consistent among views. Note that, in addition to colors, our synthesis procedure also produces the depth value for each pixel in every view in the same way we synthesize colors. Thus, a 4D patch is a 4D sub-volume in the $u-v-x-y$ space and for each point in the patch, we have its R, G, B and D values. For each point $\mathbf{p} = (\tilde{u}, \tilde{v}, \tilde{x}, \tilde{y})$ in the patch Q, we have its color $\tilde{r} = \mathbf{L}(\tilde{u}, \tilde{v}, \tilde{x}, \tilde{y})$ and depth $\tilde{d} = \mathbf{D}(\tilde{u}, \tilde{v}, \tilde{x}, \tilde{y})$. According to the depth value $\tilde{d}$, we can find its corresponding pixels in other views and collect them into the following set of colors, Γ:

$$\begin{aligned}\Gamma(\tilde{u}, \tilde{v}, \tilde{x}, \tilde{y}) = &\{\mathbf{E}_{\tilde{v}\tilde{y}}(\tilde{u}, \tilde{x} + k\tilde{d}) | k \in \{-1, +1\}\} \\ &\cup \{\mathbf{E}_{\tilde{u}\tilde{x}}(\tilde{v}, \tilde{y} + k\tilde{d}) | k \in \{-1, +1\}\}.\end{aligned} \tag{9}$$

Note that k ranges from -1 to +1 because of the patch size we used is $3 \times 3 \times 7 \times 7$. If the patch is consistent, then the set of color samples must be consistent to $\tilde{r}$. Thus, the consistency of a 4D patch Q can be measured as

$$Consistency(Q) = \sum_{\mathbf{p} \in Q} \sum_{\gamma \in \Gamma(\mathbf{p})} \psi(\gamma - \mathbf{L}(\mathbf{p})), \tag{10}$$

and the consistency of a lightfield image is defined as

$$D_{consistency}(\mathbf{T}) = \sum_{Q \in \mathbf{T}} Consistency(Q). \tag{11}$$

With this term, for our applications, given the source lightfield image $\mathbf{S}$, we find the target lightfield image $\mathbf{T}$ which optimizes the following energy with additional constraints specific to the target application:

$$\begin{aligned}\arg\min_{\mathbf{T}} &D_{completeness}(\mathbf{S}, \mathbf{T}) + D_{coherence}(\mathbf{S}, \mathbf{T}) \\ &+ D_{consistency}(\mathbf{T}).\end{aligned} \tag{12}$$

4D PatchMatch. For the optimization, the PatchMatch algorithm needs to be extended to 4D. Specifically, during the propagation and search phase, we consider 4D neighbors rather than 2D neighbors.

4. APPLICATIONS

In this section, we show a few lightfield image editing applications using the 4D patch synthesis framework described in Section 3, including completion, retargeting and reshuffling. The lightfield images were captured using a Lytro camera except for the bug example in Figure 1 which is from HCI lightfield repository. The lightfield images are of 7×7 angular resolution and 512×512 spatial resolution. With our current unoptimized implementation, the editing process in general took tens of minutes. The PatchMatch process is the most time-consuming part and it can be greatly accelerated by porting to GPUs.

Completion. For lightfield image completion, given an input lightfield image $\mathbf{S}$, the user first specifies a region to be removed in one of the input views, say $(\tilde{u}, \tilde{v})$. Our method automatically transfers the specified region to other views. Then, it removes the specified and transferred regions from all views and fills in the holes due to content removal.

Denote the removal region specified by the user as $\Omega_{\tilde{u}\tilde{v}}$. The first step is to transfer the region from the view $(\tilde{u}, \tilde{v})$ to all other views. For this purpose, we first detect SIFT features [17] within the bounding box of $\Omega_{\tilde{u}\tilde{v}}$ for the view $\mathbf{I}_{\tilde{u}\tilde{v}}$. The set of detected features is denoted as $\mathbf{F}_{\tilde{u}\tilde{v}}$. We then detect SIFT features $\mathbf{F}_{uv}$ for all views. To transfer $\Omega_{\tilde{u}\tilde{v}}$ from the view $(\tilde{u}, \tilde{v})$ to another view (u, v), we match features between $\mathbf{F}_{\tilde{u}\tilde{v}}$ and $\mathbf{F}_{uv}$. The matching is robustified by RANSAC. From the matched features, we find a homography matrix $\mathbf{H}_{uv}$. By multiplying the region $\Omega_{\tilde{u}\tilde{v}}$ with $\mathbf{H}_{uv}$, we obtain the transferred region Ω_{uv} for the view (u, v). This simple approach works well because the views in a lightfield image are usually very close. Figure 2 shows an example of region transfer.

In the application of image completion, we do not need the completeness term and only include the coherence and consistency terms. The target lightfield image $\mathbf{T}$ is initialized as the input $\mathbf{S}$ excluding the parts covered by the holes Ω_{uv}. For pixels in those parts, they are initialized as random colors. Figure 3 gives the completion results at views $(1, 1)$ and $(1, 5)$ for the example in Figure 2. Figure 3(b) gives the results using independent view synthesis. Notice that the inpainted regions on the pear are not consistent among views. It is clear that the consistency among views is not maintained this way. Figure 3(c) gives the results with the proposed method and the view consistency is maintained much better. Figure 4 shows another example of lightfield image completion.

Retargeting. The image retargeting operation changes the size of the input image while preserving its content and perception as much as possible. Given the source image $\mathbf{S}$, we initialize the target image $\mathbf{T}$ with the target size. We then use Equation (12) as the energy function to solve for $\mathbf{T}$. To avoid being trapped at local optimum, we adapted a coarse-to-fine strategy for this application. We built Gaussian image pyramids for all images in $\mathbf{S}$ and $\mathbf{T}$. Starting from the coarsest

level, we resize the image at the current level and use the result to initialize the image at the next level until reaching the finest level. Figure 5 gives an example of lightfield image retargeting in which we reduced the width by 15%.
Reshuffling. In this application, the user first specifies how to rearrange the layout of the image and the system synthesizes images to minimize Equation (12) while satisfying the specified constraints. The coarse-to-fine strategy is also adapted in this application. Figure 6 shows two examples of lightfield image reshuffling. In the first example, we duplicated the lamppost. In the second example, we added a penthouse on the roof.

5. CONCLUSIONS

We have presented a lightfield patch-based synthesis framework which handles the corresponding information in all views and jointly synthesizes contents with a consistent scene interpretation. The combination of the patch consistency metric and the joint 4D patch search contributes to the realism and consistency of the synthesized lightfield images and plausibility of their scene interpretation. The method has potential to be useful for many lightfield editing processing applications as demonstrated in the experiments.

6. REFERENCES

[1] Antonio Criminisi, Patrick Perez, and Kentaro Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, September 2004.

[2] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani, “Summarizing visual data using bidirectional similarity,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[3] Yael Pritch, Eitam Kav-Venaki, and Shmuel Peleg, “Shift-map image editing,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, September 2009, pp. 151–158.

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman, “PatchMatch: a randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 24:1–24:11, 2009.

[5] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B. Goldman, and Pradeep Sen, “Image melding: Combining inconsistent images using patch-based synthesis,” *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 82:1–82:10, July 2012.

[6] Liang Wang, Hailin Jin, Ruigang Yang, and Minglun Gong, “Stereoscopic inpainting: Joint color and depth completion from stereo images,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[7] B. Morse, J. Howard, S. Cohen, and B. Price, “Patchmatch-based content completion of stereo image pairs,” in *Proceedings of 3DIMPVT*, 2012, pp. 555–562.

[8] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross, “Nonlinear disparity mapping for stereoscopic 3D,” *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 75:1–75:10, 2010.

[9] Che-Han Chang, Chia-Kai Liang, and Yung-Yu Chuang, “Content-aware display adaptation and interactive editing for stereoscopic images,” *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 589–601, 2011.

[10] Ken-Yi Lee, Cheng-Da Chung, and Yung-Yu Chuang, “Scene warping: Layer-based stereoscopic image resizing,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 49–56.

[11] Sheng-Jie Luo, Ying-Tse Sun, I-Chao Shen, Bing-Yu Chen, and Yung-Yu Chuang, “Geometrically consistent stereoscopic image editing using patch-based synthesis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 1, pp. 56–67, January 2015.

[12] Sven Wanner and Bastian Goldluecke, “Globally consistent depth labeling of 4d light fields,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 41–48.

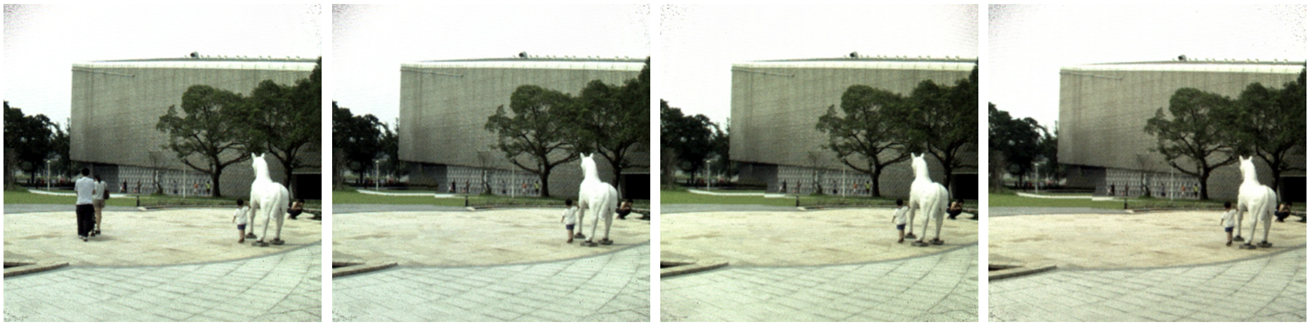
[13] Sven Wanner, Christoph Straehle, and Bastian Goldluecke, “Globally consistent multi-label assignment on the ray space of 4d light fields,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1011–1018.

[14] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus Gross, “Scene reconstruction from high spatio-angular resolution light fields,” *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 73:1–73:12, July 2013.

[15] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu, “Saliency detection on light field,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

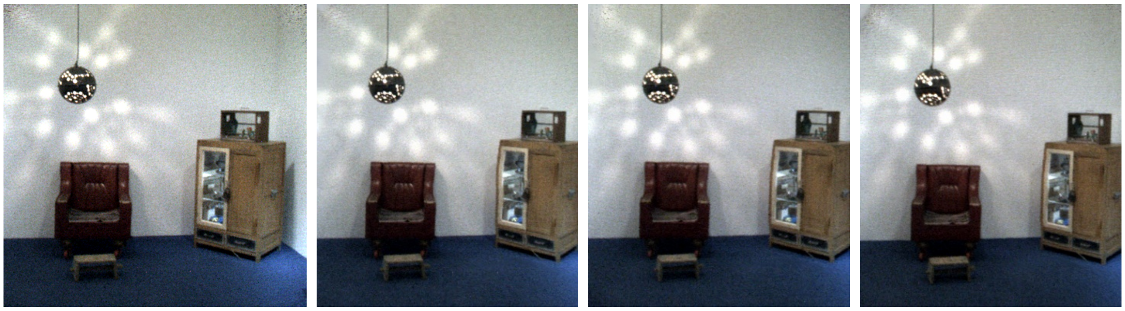
[16] C. Birklbauer and O. Bimber, “Light-field retargeting,” *Computer Graphics Forum*, vol. 31, no. 2pt1, pp. 295–303, May 2012.

[17] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.



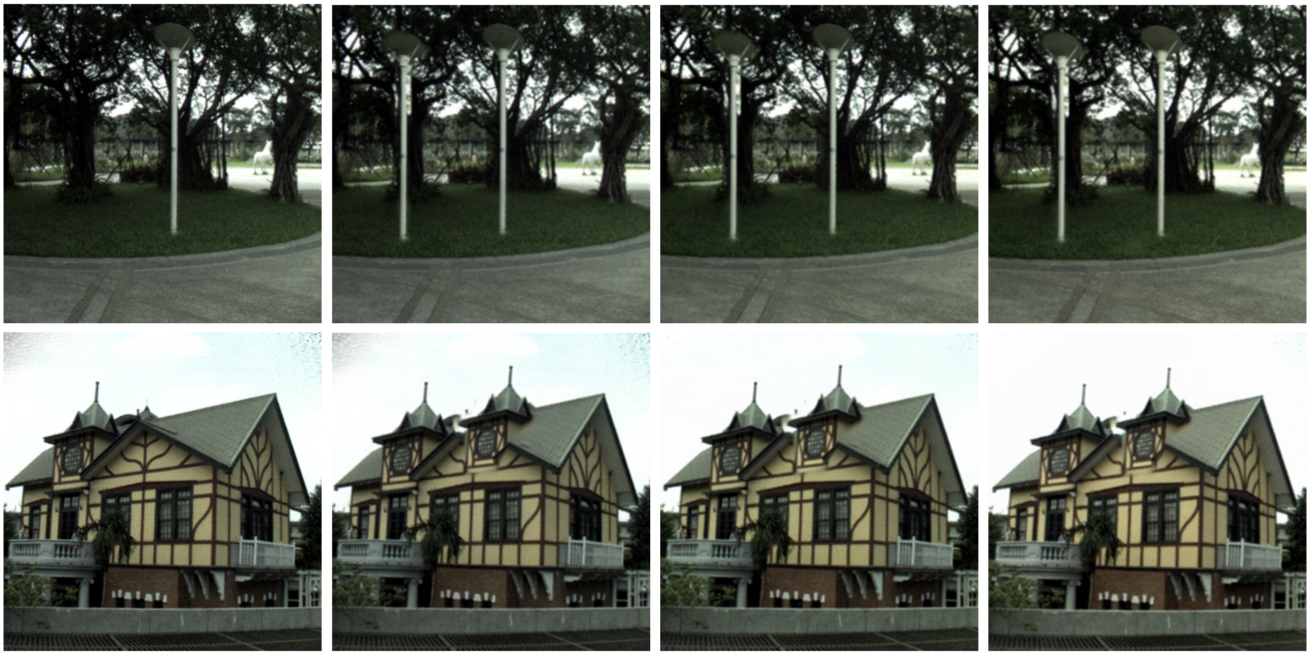
input view (3, 3) synthesized view (3, 3) synthesized view (7, 3) synthesized view (3, 7)

Fig. 4: The results of lightfield image completion.



input view (3, 3) synthesized view (3, 3) synthesized view (7, 3) synthesized view (3, 7)

Fig. 5: The results of lightfield image retargeting.



input view (4, 4) synthesized view (4, 4) synthesized view (7, 4) synthesized view (4, 7)

Fig. 6: The results of lightfield image reshuffling. In the first example (top row), we duplicate the lamppost in the input. In the second example (bottom role), we modify the structure of the house by adding another penthouse.