

# Light Field Editing Based on Reparameterization

Hongbo Ao<sup>1</sup>, Yongbing Zhang<sup>1</sup>(✉), Adrian Jarabo<sup>3</sup>, Belen Masia<sup>3,4</sup>,  
Yebin Liu<sup>2</sup>, Diego Gutierrez<sup>3</sup>, and Qionghai Dai<sup>2</sup>

<sup>1</sup> Graduate School at Shenzhen, Tsinghua University, Shenzhen, China  
zhang.yongbing@sz.tsinghua.edu.cn

<sup>2</sup> Department of Automation, Tsinghua University, Beijing, China

<sup>3</sup> Universidad de Zaragoza, Zaragoza, Spain

<sup>4</sup> MPI Informatik, Saarbrücken, Germany

**Abstract.** Edit propagation algorithms are a powerful tool for performing complex edits with a few coarse strokes. However, current methods fail when dealing with light fields, since these methods do not account for view-consistency and due to the large size of data that needs to be handled. In this work we propose a new scalable algorithm for light field edit propagation, based on reparametrizing the input light field so that the coherence in the angular domain of the edits is preserved. Then, we handle the large size and dimensionality of the light field by using a downsampling-upsampling approach, where the edits are propagated in a reduced version of the light field, and then upsampled to the original resolution. We demonstrate that our method improves angular consistency in several experimental results.

**Keywords:** Light field · Edit propagation · Reparameterization · Clustering

## 1 Introduction

In the last years, light fields [10] have gained attention as a plausible alternative to traditional photography, due to its increased post-processing capabilities, including refocus, view shifting or depth reconstruction. Moreover, both plenoptic cameras (e.g. *Lytro*<sup>TM</sup> or *Raytrix*<sup>TM</sup>) or automultiscopic displays [13] using light fields have appeared in the consumer market. The wide-spread of this data has created a new need for providing similar manipulation capabilities of traditional images or videos. However, only a few seminal works [6,7] have been proposed to fill this gap.

Editing a light field is challenging for two main reasons: (i) the increased dimensionality and size of the light field makes it harder to efficiently edit it, since the edits need to be performed in the full dataset; and (ii) angular coherence needs to be preserved to provide an artifact free solution. In this work we propose a new technique to effectively edit light fields based on propagating the edits

specified in a few sparse coarse strokes. The key idea of our method is to include a novel light field reparametrization that allows us to implicitly impose view-coherence in the edits. Then, inspired in the work by Jarabo et al. [7], we propose a downsampling-upsampling approach, where the edit propagation routine is done in a significantly reduced dataset, and then the result is upsampled to the full-resolution light field.

In comparison to previous work, our results *preserve view-coherence* thanks to the reparametrization of the light field, is *scalable* in both time and memory and is easy to implement on top of any propagation machinery.

## 2 Related Work

Previous works mainly focus on edit propagation on single images, with some extensions to video. Levin et al. [9] formulate a local optimization to propagate user scribbles to the expected regions in the target image. The method requires a large set of scribbles or very large neighborhoods to propagate the edits in the full image. In contrast, An and Pellacini [1] propose a global optimization algorithm by considering similarity between all the possible pixel pairs in a target image; they formulate propagation as a quadratic system and solve it efficiently by taking advantage of its low-rank nature. However, this method scales linearly with the size of the problem, and does not account for view coherence. Xu et al. [15] improve An and Pellacini’s method by downsampling the data using a kd-tree in the affinity space, which allows them handling large datasets. However, they scale poorly with the number of dimensions.

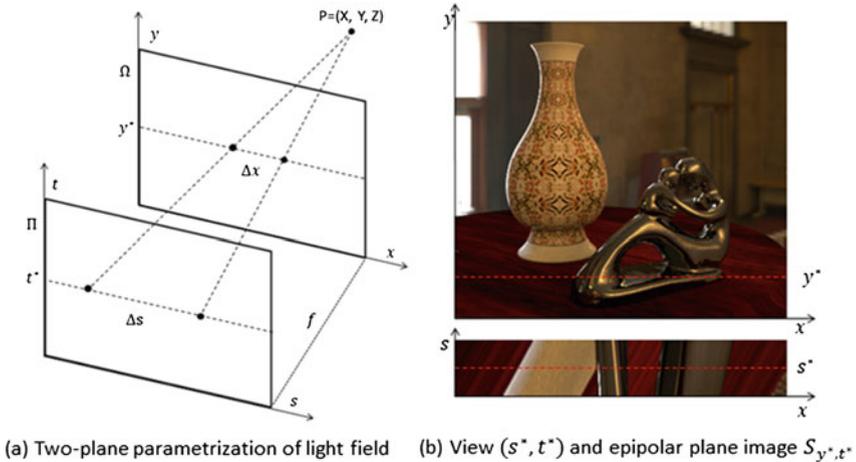
Other methods propose to increase the efficiency and generality of the propagation by posing as different energy minimization systems: Li et al. [11] reformulate the propagation problem as an interpolation problem in a high-dimensional space, which could be solved very efficiently using radial basis functions. Chen et al. [5] design a manifold preserving edit propagation algorithm, based on the simple intuition that each pixel in the image is a linear combination of other pixels which are most similar with the target pixel. The same authors later improve this work by propagating first in the basis of a trained dictionary, which is later used to reconstruct the final image [4]. Xu et al. [16] derive a sparse control model to propagate sparse user scribbles successfully to all the expected pixels in the target image. Finally, Ao et al. [2] devise a hybrid domain transform filter to propagate user scribbles in the target image. None of these works are designed to work efficiently with the high-dimensional data of light fields, and might produce inconsistent results between views, that our light field reparametrization avoids.

Finally, Jarabo et al. [7] propose a novel downsampling-upsampling propagation method, which handles the high dimensionality of light fields. We solve our problem efficiently inspired by their approach, although they do not enforce view consistency. This is to the best of our knowledge the only work dealing with edit propagation in light fields, while most previous effort on light field editing have focused on local edits [6, 12, 14] or light field morphing [3, 18].

### 3 Light Field Editing Framework

The proposed algorithm can be divided into two parts. The first part is light field reparameterization, while the other one is the downsampling-upsampling propagation framework. The latter can be split into three phases: downsampling the light field, propagation on the downsampled light field, and guided upsampling of the propagated data.

We rely on the well-known two-plane parameterization of a light field [10], shown in Fig. 1 (a), in which each ray of light  $\mathbf{r}$  in the scene can be defined as a 4D vector which codes its intersection with each of the two planes  $\mathbf{r} = [s, t, x, y]$ . One of the planes can be seen as the camera plane, where the cameras are located (plane  $st$ ), and the other as the focal plane (plane  $xy$ ). Note that the radiance can be reduced to a 4D vector because we assume it travels through free space (and thus does not change along the ray). It is often beneficial to look at the epipolar plane images of the light field. An epipolar volume can be built by stacking the images corresponding to different viewpoints; once this is done, if we fix e.g. the vertical spatial coordinate along the volume, we can obtain an epipolar image or EPI (Fig. 1 (b)).



**Fig. 1.** (a) Two-plane parametrization of a light field. Plane  $\Pi$  represents the camera plane of the light field, plane  $\Omega$  represents the focal plane. Each camera location  $(s^*, t^*)$  yields a different view of the scene. (b) A view located at  $(s^*, t^*)$  and epipolar image  $S_{y^*, t^*}$ . We can obtain an epipolar image by fixing a horizontal line of constant  $y^*$  in the focal plane  $\Omega$  and a constant camera coordinate  $t^*$  in the camera plane  $\Pi$ .

Once we model the light field with the two-plane parameterization, each pixel in the light field can be characterized by an 8D vector when color and depth information are taken into account. We thus express each pixel  $\mathbf{p}$  in the light field as a 8D vector  $\mathbf{p} = [r, g, b, x, y, s, t, d]$ , where  $(r, g, b)$  are the colors of the

pixel,  $(x, y)$  are the image coordinates on plane  $\Omega$ ,  $(s, t)$  are the view coordinates on plane  $\Pi$  and  $d$  is the depth information of the pixel. This notation will be used throughout the rest of the paper.

### 3.1 Light Field Reparameterization

One of the main challenges when doing light field editing is preserving view consistency. Each object point in the light field has a corresponding image point in each of the views of it (excepting occlusions), and these follow a slanted line (with slant related to the depth of the object) in the epipolar images. Here, we exploit this particular structure of epipolar images and propose a well-designed transformation of the light field data that will help preserve this view consistency when performing editing operations.

This transformation amounts to reparameterizing the light field by assigning to each pixel  $\mathbf{p}$  a transformed set of xy coordinates,  $(x', y')$ , such that the pixel, in the transformed light field, will be defined by vector  $[r, g, b, x', y', s, t, d]$ . These new coordinates, will result in a transformed light field in which pixels corresponding to the same object point will be vertically aligned in the epipolar image, that is, will not exhibit spatial variation with the angular dimension; this process is illustrated in Fig. 2, which shows an original epipolar image and the same image after re-parameterization.

The actual computation of these new coordinates is given by Eqs. 1 and 2:

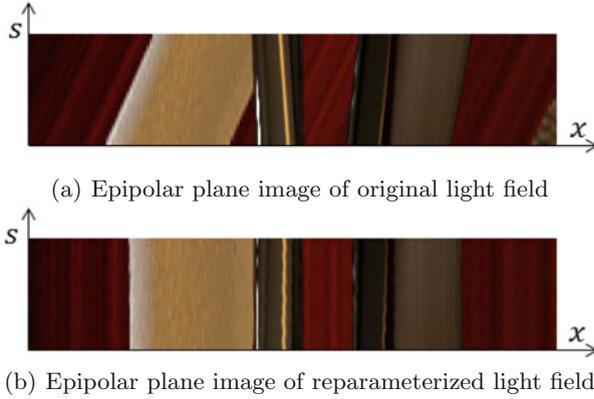
$$x' = \psi(x, y, d) = x - (y - y_c) \cdot (d - 1), \quad (1)$$

$$y' = \phi(x, y, d) = y - (x - x_c) \cdot (d - 1), \quad (2)$$

where  $x_c$  and  $y_c$  are the coordinates of the middle row and middle column of the epipolar images, respectively, in order to set the origin at the center, and  $d$  is, as mentioned, the depth information of that pixel. Note that the reparameterization can be applied to both the  $y-t$  slices and the  $x-s$  slices of the light field. Using this simple transformation will help in maintaining view consistency within the light field data.

### 3.2 Downsampling-Upsampling Propagation Framework

To efficiently address the propagation task, we build on the downsampling and upsampling propagation framework proposed by Jarabo et al. [7]. The improved downsampling-upsampling framework implements a three-step strategy to propagate scribbles on the reparameterized light field. To enable efficient calculation, the downsampling-upsampling propagation framework first makes use of k-means clustering algorithm [17] to downsample the light field data in the 8D space. Then a global optimization-based propagation algorithm is applied to the downsampled light field data. Finally, a joint bilateral upsampling method is used to interpolate the propagated data to the resolution of the original light field.



**Fig. 2.** (a) Epipolar image  $S_{y^*,t^*}$  of the original light field. (b) Reparameterized epipolar image  $S'_{y^*,t^*}$ .

**Downsampling Phase.** To dispose of the unacceptable poor propagation efficiency due to the extremely large size of the light field data, and taking advantage of the large redundancy in it, we use k-means clustering [17] to downsample the original light field data to a smaller size data set. The downsampling phase successfully decreases the data redundancy by representing all the pixels in one cluster with the corresponding cluster center.

Given the original light field data we cluster the  $M \times N^1$  8D data points into  $K$  clusters ( $K \ll N$ ), and thus merely need to propagate within the  $K$  cluster center points. Each cluster is denoted by  $C_k$ ,  $k \in [1, 2, 3, \dots, K]$ , and each cluster center is expressed as  $\mathbf{c}_k$ ,  $k \in [1, 2, 3, \dots, K]$ . The set  $\mathbf{c}_k$ ,  $k \in [1, 2, 3, \dots, K]$ , is therefore the downsampled light field.

Original scribbles drawn by the user to indicate the edits to be performed also need to be downsampled according to the cluster results. A weight matrix  $\mathbf{D} \in \mathbb{R}^{M \times N}$  is used to record which pixel in the original light field is covered with user scribbles by setting the corresponding element in the matrix to 1 where a user scribble is present, and otherwise set the corresponding element to 0. Assume the original scribbles are expressed as  $\mathbf{S} \in \mathbb{R}^{M \times N}$ , then the new scribbles of the downsampled light field  $\mathbf{s}_k$  can be calculated as follows:

$$\mathbf{s}_k = \frac{1}{M_0} \sum_{(i,j) \in \{(m,n) | \mathbf{p}_{mn} \in C_k\}} D_{ij} * S_{ij}, \tag{3}$$

$$M_0 = \sum_{(i,j) \in \{(m,n) | \mathbf{p}_{mn} \in C_k\}} D_{ij}, \tag{4}$$

where  $\mathbf{p}_{mn}$ ,  $m = [1, 2, \dots, M]$ ,  $n = [1, 2, \dots, N]$  are 8D pixel vectors in the original light field. We get the downsampled scribble set  $\{\mathbf{s}_k\}$ ,  $k = [1, 2, \dots, K]$ , according

<sup>1</sup> We can represent the light field as a 2D matrix composed of the different views, as per common practice.

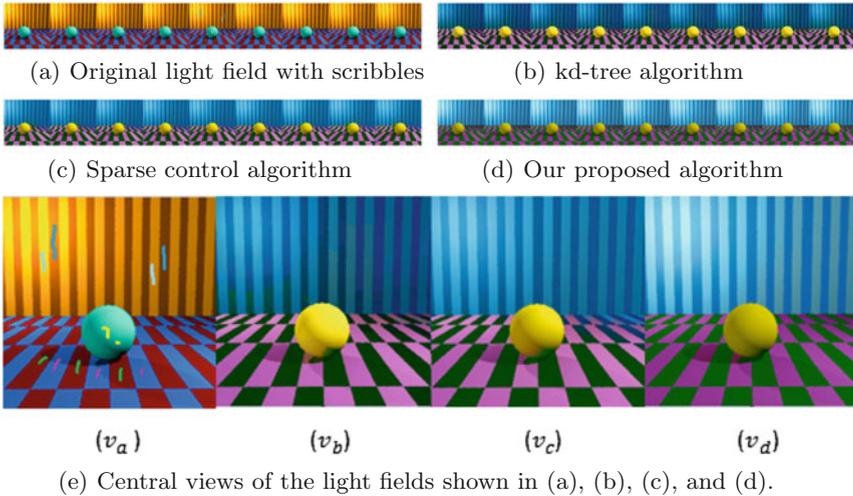
to Eqs. 3 and 4. Considering the redundancy of light field data, a small value  $K$  will be good enough to downsample the original light field.

**Propagation Phase.** After the downsampling phase, we get the downsampled light field data  $\mathbf{c}_k$ ,  $k \in [1, 2, 3, \dots, K]$  and its corresponding scribble set  $\{\mathbf{s}_k\}$ . We adopt the optimization framework proposed by An and Pellacini [1] to propagate scribbles  $\mathbf{s}_k$  on the new light field data  $c_k$ . We formulate the propagation algorithm in Eqs. 5 and 6, and by optimizing this expression we can acquire the propagated result  $\mathbf{e}_k$ .

$$\sum_k \sum_j \omega_j z_{kj} (\mathbf{e}_k - \mathbf{s}_j)^2 + \lambda \sum_k \sum_j z_{kj} (\mathbf{e}_k - \mathbf{e}_j)^2, \tag{5}$$

$$z_{kj} = \exp(-\|(\mathbf{c}_k - \mathbf{c}_j) \cdot \boldsymbol{\sigma}\|_2^2), \tag{6}$$

where  $\mathbf{c}_k = (r_k, g_k, b_k, x_k, y_k, s_k, t_k, d_k)$  is pixel vector of the new light field  $\mathbf{c}_k$ ,  $k \in [1, 2, 3, \dots, K]$ ;  $z_{kj}$  is the similarity between pixel vectors  $k$  and  $j$ ;  $\boldsymbol{\sigma} = (\sigma_c, \sigma_c, \sigma_c, \sigma_i, \sigma_i, \sigma_v, \sigma_v, \sigma_d)$  are the weights of each feature in the 8D vector used to compute the affinity and thus to determine the extent of the propagation in those dimensions; and  $\omega_j$  is a weight coefficient which is set to 1 when  $s_j$  is not zero and is otherwise set to 0. For a small number of cluster centers, i.e. a small  $K$ , Eq. 5 can be solved efficiently.



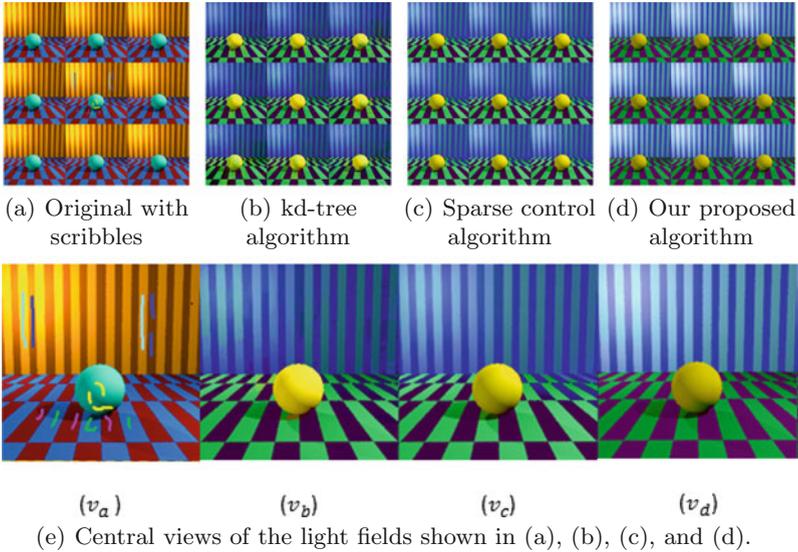
**Fig. 3.** Light field editing result on a 3D light field (horizontal parallax only). We show the initial scribbles drawn by the user and our result compared to that of two other algorithms. Note that (e) shows the central views of the different light fields shown, where the differences can be appreciated. Please refer to the text for details.

**Upsampling Phase.** Finally, we need to calculate the edited result of all the pixels in the original data set. In the upsampling phase, we utilize the propagated result set  $\mathbf{e}_k$  to obtain the resulting appearance of each pixel in the full light field.

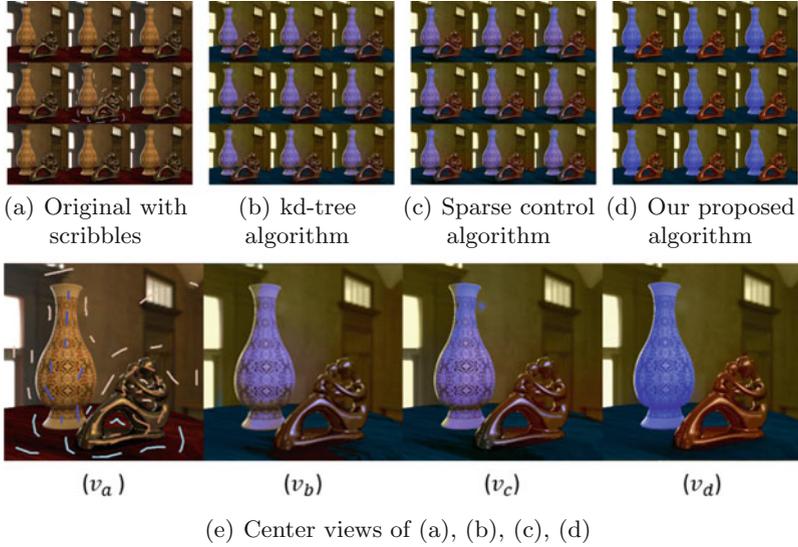
For each pixel in the original light field, we find  $n$  nearest neighbor cluster centers in the downsampled light field data set  $\mathbf{c}_k$  by using a kd-tree for the searching process. Each pixel  $\mathbf{p}$  will relate to one nearest neighbor cluster set  $\Omega = \{c_j, j = 1, 2, 3, \dots, m\}$  after the nearest neighbor search procedure. Then joint bilateral upsampling [8] will be used in the upsampling process. More formally, for an arbitrary pixel position  $\mathbf{p}$ , the filtered result can be formulated as:

$$E(\mathbf{p}) = \frac{1}{k_p} \sum_{q \downarrow \in \Omega} e_{q \downarrow} f(\|\mathbf{p} \downarrow - \mathbf{q} \downarrow\|) g(\|\mathbf{I}_p - \mathbf{I}_q\|), \quad (7)$$

where  $f(x)$  and  $g(x)$  are exponential functions ( $\exp(x)$ );  $q \downarrow$  and  $p \downarrow$  are the positional coordinates of the downsampled light field;  $e_{q \downarrow}$  is the color of the pixel vector in propagated light field;  $\mathbf{I}_p$  and  $\mathbf{I}_q$  are the pixel vectors in the original light field; and  $k_p$  is a normalizing factor, which is the sum of the  $f \cdot g$  filter weights.



**Fig. 4.** Light field editing result on a 4D light field. We show the initial scribbles drawn by the user and our result compared to that of two other algorithms. Note that (e) shows the central views of the different light fields shown, where the differences can be appreciated. Please refer to the text for details.



**Fig. 5.** Another light field editing result on a more complex 4D light field. We show the initial scribbles drawn by the user and our result compared to that of two other algorithms. Note that (e) shows the central views of the different light fields shown, where the differences can be appreciated. Please refer to the text for details.

## 4 Results

In this section we show our results and compare with two state-of-the-art edit propagation algorithms: a kd-tree based method [15], and a sparse control method [16]. In the result shown in Fig. 3, we recolor the light field propagating a few scribbles on the center view of a  $1 \times 9$  horizontal light field. We show the original light field with user scribbles on the center view, the results of the two previous methods, and our own, as well as larger center views for all for easier visual analysis. Our algorithm ( $v_d$ ) preserves the intended color of the input scribbles better results, while avoids artifacts such as color bleeding into different areas. In contrast, both the kd-tree ( $v_b$ ) and sparse control ( $v_c$ ) methods produce some blending between the colors of the wall and the floor. This blending is also responsible on the change of the user-specified colors, which are darker in  $v_b$  and  $v_c$ , that our method propagates more faithfully.

In Figs. 4 and 5, we draw some scribbles on the center view of a  $3 \times 3$  light field. Again, we show the input scribbles, a comparison between the previous methods and ours, plus larger central views. Similar to the results in Fig. 3, our method propagates more faithfully the input colors from the user. In addition, our method results into proper color segmentation based on the affinity of the different areas of the light field, while the results of the kd-tree ( $v_b$ ) and sparse control methods ( $v_c$ ) exhibit clear artifacts in form of blended colors, or wrongly propagated areas.

## 5 Conclusion

We have presented a light field edit propagation algorithm, based on a simple re-parameterization that aims to better preserve consistency between the edited views. We have incorporated it into a downsampling-upsampling framework [7], which allows to handle efficiently the large amounts of data that describe a light field. Our initial results show improvements over other existing edit propagation methods. These are the first steps in a possible direction towards the long-standing goal of multidimensional image editing. Further analysis and developments are needed to exhaustively test the validity of the approach.

**Acknowledgements.** The project is supported by the National key foundation for exploring scientific instrument No. 2013YQ140517 and partially supported by the National Natural Science Foundation of China under Grants 61170195, U1201255 & U1301257, the Spanish Ministry of Science and Technology (project LIGHTSLICE) and the BBVA Foundation. Diego Gutierrez is additionally supported by a Google Faculty Research Award. Belen Masia is partially supported by the Max Planck Center for Visual Computing and Communication.

## References

1. An, X., Pellacini, F.: Approp: all-pairs appearance-space edit propagation. *ACM Trans. Graph. (TOG)* **27**(3), 40 (2008)
2. Ao, H., Zhang, Y., Dai, Q.: Image colorization using hybrid domain transform. In: ICASSP, January 2015
3. Chen, B., Ofek, E., Shum, H.Y., Levoy, M.: Interactive deformation of light fields. In: Proceedings of the I3D 2005, pp. 139–146 (2005)
4. Chen, X., Zou, D., Li, J., Cao, X., Zhao, Q., Zhang, H.: Sparse dictionary learning for edit propagation of high-resolution images. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2854–2861. IEEE (2014)
5. Chen, X., Zou, D., Zhao, Q., Tan, P.: Manifold preserving edit propagation. *ACM Trans. Graph. (TOG)* **31**(6), 132 (2012)
6. Jarabo, A., Masia, B., Bousseau, A., Pellacini, F., Gutierrez, D.: How do people edit light fields? *ACM Trans. Graph.* **33**(4), 146:1–146:10 (2014)
7. Jarabo, A., Masia, B., Gutierrez, D.: Efficient propagation of light field edits. In: Proceedings of the SIACG 2011 (2011)
8. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. *ACM Trans. Graph. (TOG)* **26**, 96 (2007). ACM
9. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. *ACM Trans. Graph. (TOG)* **23**, 689–694 (2004). ACM
10. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 31–42. ACM (1996)
11. Li, Y., Ju, T., Hu, S.M.: Instant propagation of sparse edits on images and videos. In: *Computer Graphics Forum*, vol. 29, pp. 2049–2054. Wiley Online Library (2010)
12. Masia, B., Jarabo, A., Gutierrez, D.: Favored workflows in light field editing. In: CGVCVIP (2014)

13. Masia, B., Wetzstein, G., Didyk, P., Gutierrez, D.: A survey on computational displays: pushing the boundaries of optics, computation and perception. *Comput. Graph.* **37**, 1012–1038 (2013)
14. Seitz, S.M., Kutulakos, K.N.: Plenoptic image editing. *Int. J. Comput. Vision* **48**(2), 115–129 (2002)
15. Xu, K., Li, Y., Ju, T., Hu, S.M., Liu, T.Q.: Efficient affinity-based edit propagation using kd tree. *ACM Trans. Graph. (TOG)* **28**, 118 (2009). ACM
16. Xu, L., Yan, Q., Jia, J.: A sparse control model for image and video editing. *ACM Trans. Graph. (TOG)* **32**(6), 197 (2013)
17. Žalik, K.R.: An efficient k-means clustering algorithm. *Pattern Recogn. Lett.* **29**(9), 1385–1391 (2008)
18. Zhang, Z., Wang, L., Guo, B., Shum, H.Y.: Feature-based light field morphing. *ACM Trans. Graph.* **21**(3) (2002). <http://doi.acm.org/10.1145/566654.566602>