# 4D Light Field Segmentation with Spatial and Angular Consistencies

Hajime Mihara[1] Takuya Funatomi[1] Kenichiro Tanaka[2,1] Hiroyuki Kubo[1]
Hajime Nagahara[3] Yasuhiro Mukaigawa[1]
[1]Nara Institute of Science and Technology [2]Osaka University [3]Kyushu University

http://omilab.naist.jp/project/LFseg/

## Abstract

*In this paper, we describe a supervised four-dimensional (4D) light field segmentation method that uses a graph-cut algorithm. Since 4D light field data has implicit depth information and contains redundancy, it differs from simple 4D hyper-volume. In order to preserve redundancy, we define two neighboring ray types (spatial and angular) in light field data. To obtain higher segmentation accuracy, we also design a learning-based likelihood, called objectness, which utilizes appearance and disparity cues. We show the effectiveness of our method via numerical evaluation and some light field editing applications using both synthetic and real-world light fields.*

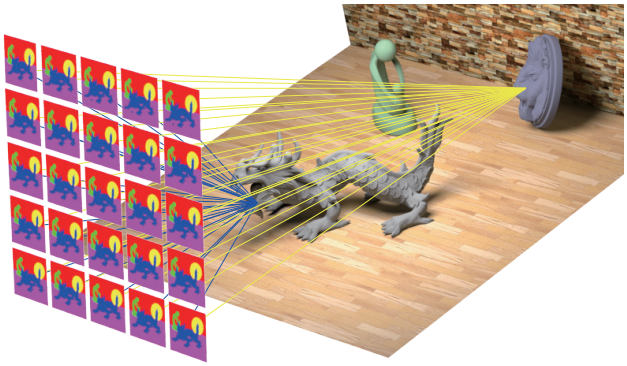


Figure 1: Our method assigns label to each ray considering the geometrical structure of a light field.

## 1. Introduction

Light field imaging is spreading as a result of recent developments in light field acquisition systems [3, 20, 22] and light field cameras are expected to take the place of conventional digital cameras in the near future. As light field photography has increased in popularity, demands for light field editing methods and tools have increased as well, in the same manner as products such as Photoshop® have become very popular editing tools for pictures. For example, Horn and Chen [13] have proposed a system that allows users to interactively manipulate, composite, and render multiple light fields.

However, Jarabo *et al*. [14] shows that light field editing remains a challenging task for the following reasons: 1) a light field is a four-dimensional (4D) data structure, while the majority of displays and input devices today are designed for two-dimensional (2D) content. 2) A light field is redundant, which implies that any local edit on a light field needs to be propagated coherently to preserve this redundancy, and 3) while a light field provides a vivid sense of depth, this depth information is not encoded explicitly.

While an effective light field editing tool should have many of the same functions as Photoshop®, one of the most fundamental functions is region selection. As a method of achieving efficient region selection in a 4D light field, we propose a multi-label segmentation framework that is capable of coping with the difficulties mentioned above.

Our framework assumes that the user will select a portion of the regions contained in the 4D light field data via a 2D input device. Although functional user interface (UI) for this purpose is proposed by Jarabo *et al*. [14], it remains difficult for users to manually select an entire region in 4D space. In our method, the user needs to provide the cues to specify a region of interest by inputting a label on a portion of the region. For example, the user can use brush strokes to input a label on the center viewpoint image, which is a subspace of the light field data, in order to specify the target object to be edited. Based on those user inputs, our algorithm identifies the appropriate regions in the 4D light field.

One may consider that segmentation results for a 4D light field can be obtained by applying a 2D image segmentation method to each viewpoint image, there is no guarantee that the redundancy of the 4D light field is preserved. This is because, even though the light field data has a 4D structure, neighboring relationships are not as simple as

those of a regular three-dimensional (3D) volume grid, primarily due to redundancy complications. To compensate for this redundancy, we utilize disparities in 4D light field to define appropriate neighboring relationships in order to preserve redundancies. Additionally, we define neighboring relationships among the rays in the 4D light field data in order to estimate the regions of interest coherently.

Although light field data does not have explicit depth information, some methods such as [26] have been proposed to exploit its disparities. Our framework utilizes color as well as disparity information to estimate regions appropriately. Because color and disparity information are inherently dissimilar, integrating them to determine the regions of interest is a difficult task. To integrate color and disparity cues, we employ a learning based approach in order to evaluate whether each ray is included in a selected region or not.

## 2. Related Work

Segmentation is one of the most important tasks in the computer vision field and has been studied for many years. One of the most famous image segmentation methods is GrabCut [25], which is a supervised foreground/background segmentation method for 2D image and has already been implemented in many image editing software applications. GrabCut is built upon graph-cut algorithms [4, 5, 6]. Graph-cuts can be applied to any type of dimensional data including imagery, video footage, and 3D structures [4], and can also be extended to multi-label segmentation [6].

In graph-cut approaches, the data are treated as a graph structure with vertices and edges. A vertex represents each pixel, and neighboring pixels are connected by an weighted edge based on their similarity. In cases involving multi-label image segmentation, each label also has a special vertex, which is called a terminal. Pixels' vertices are connected to all terminals, where their edge weights determine the likelihood of label assignment. The solution to finding the cut on the graph at minimum cost lies in obtaining the segmentation at minimum amount of energy, and this problem can be solved by min-cut/max-flow algorithm [12, 5]. After cutting, each vertex is connected to a single terminal, which means the corresponding label is assigned to the corresponding pixel. Our method also employs a graph-cut segmentation approach. These segmentation methods are categorized as supervised methods since they require user inputs in the form of clues.

Although some graph-cut methods [4, 6] can process any-dimensional data, they are not always suitable for high-dimensional data, such as video footage. Unlike 3D volume data, video data have an irregular structure along the time axis. Hence, segmentation methods can be improved by appropriately considering the irregular neighbor relationships of the data. For example, Nagahashi *et al*. [21] improve the accuracy of video segmentation by defining temporal neighbor relationships, which are corresponding pixels of neighboring frames. One problem with 4D light field data that is similar to video data is that the redundancy in a light field is complex. Our method is the first method that is capable of focusing on the 4D light field segmentation using a graph-cut approach.

Meanwhile, there are some unsupervised methods that can be used for multi-view images or 4D light field segmentation. Berent *et al*. [2] propose a 4D light field segmentation method based on a level set method [23], in which an active contour method is applied to segment a 4D hyper-volume. Kolmogorov and Zabih [15] propose a depth labeling method for multi-view images based on the fact that foreground object cannot be occluded by deeper objects. Kowdle *et al*. [16] propose a method of automatic object extraction from multi-view images using disparity cue. Their method uses appearance and disparity cues in multi-view images to determine the likelihood of foreground objects. Xu *et al*. [29] propose an unsupervised approach to localize transparent object regions in a light field image. Their method uses a light field distortion feature [19], which represents the likelihood that the pixel belongs to a transparent object region, and a binary graph-cut segmentation method. While these methods are effective, unsupervised approaches are unsuitable for region selection for light field editing because regions of interest differ from user to user.

As for supervised approaches, Wanner *et al*. [28] propose a image segmentation method using a 4D light field, which uses both appearance and disparity cues similar to [16]. To cope with these inherently different information types, they train a random forest classifier in order to integrate appearance and disparity and obtain a single likelihood for each label. While effective, their method only outputs segmentation results on the center 2D image.

We use GCDL[26] to exploit disparity from light field data, and use both appearance and disparity to evaluate the likelihood of each region for each ray. In our method, we use a support vector machines (SVMs) to integrate the likelihoods into a graph-cut algorithm.

## 3. Light field segmentation

As previously stated, the purpose of our method is to assign a label to each ray. In the following section, we show some assumptions based on light field structures, after which we will formulate the light field segmentation problem as an energy minimization problem.

### 3.1. Representation of light field

While there are various ways to represent 4D light fields [1, 11, 17], we have adopted the Lumigraph

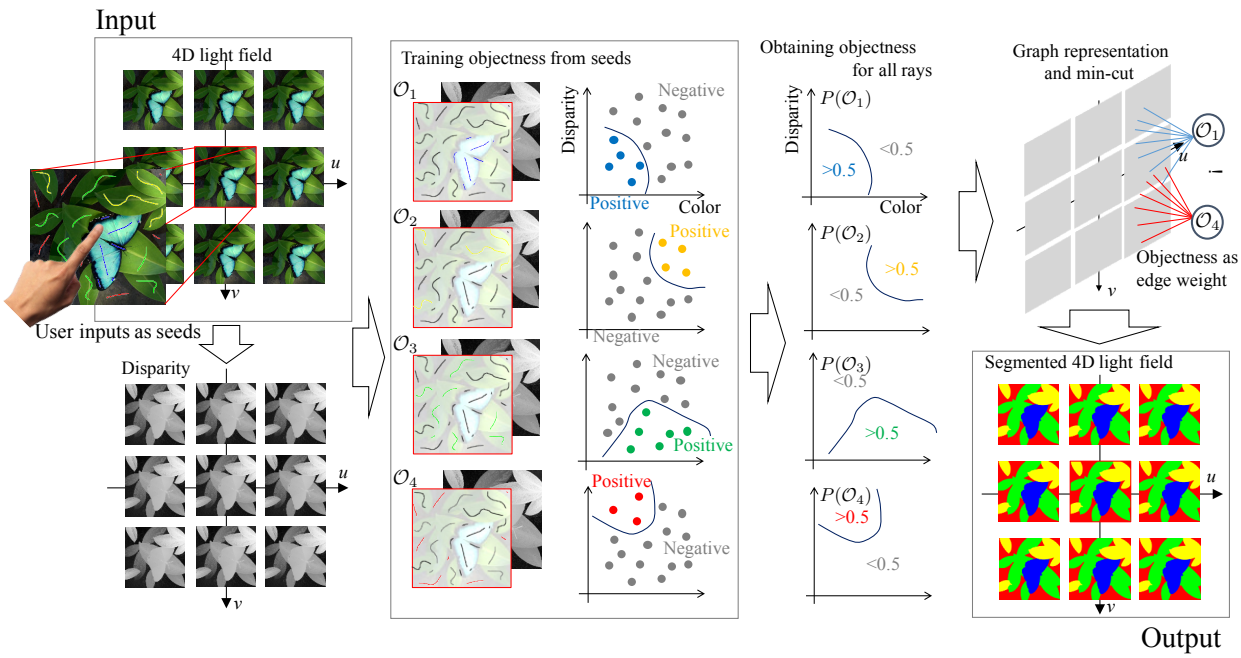
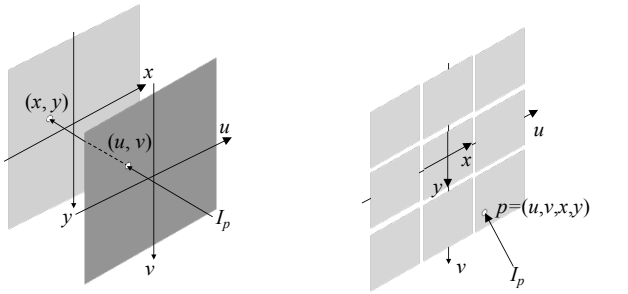



Figure 3: Overall flow of light field segmentation. The input is a 4D light field and the user specified seeds in the center viewpoint. SVMs for each label are obtained from the color and estimated disparity distribution. Objectnesses are obtained from the distance between each of the decision planes. Ultimately, the 4D light field is segmented based on a graph-cut.

method [11] to model rays in 3D space. A ray is defined by two points of intersections with the $u$–$v$ and $x$–$y$ planes in the 3D coordinate as shown in Fig. 2a. A ray can be represented as a point in 4D space as $p = (u, v, x, y)$, and the intensity of $p$ is represented as $I_p$. The Lumigraph representation can be converted into a multi-view representation [17] containing a viewpoint plane and an image plane as shown in Fig. 2b, and vice versa. In a multi-view, the $u$–$v$ and $x$–$y$ planes respectively correspond to the viewpoint and image planes. In this paper, we will explain our method using a multi-view representation to facilitate understandability.




(a) Lumigraph representation

(b) Multi-view representation

Figure 2: Two types of 4D light field representation: (a) light field representation by Lumigraph[11]. (b) Multi-view representation. These representations are essentially equivalent [17]. In this paper, we adopt a multi-view representation.

### 3.2. Framework overview

The overall light field segmentation flow is shown in Fig. 3. Here, let $\mathcal{O}_i$ be the $i$-th label $(i = 1, \cdots, n)$, where $n$ denotes the number of labels. Our goal is to assign one of the labels to each ray at $p$ based on user inputs, called *seeds*. Besides, we assume that disparity $d_p$ is known for the given light field data, because a number of methods are available for accurately estimating disparities from a light field [7, 18, 26].

Similar to [16, 28], our algorithm calculates the likelihood of each label, called *objectness*, from the intensity and disparity of the ray. To integrate these inherently different information types into one objectness measure, we use an SVM classifier. The objectness is defined as the distance from the decision hyperplane of the classifier. Additionally, we consider both the spatial and angular neighborhoods of a ray in a 4D light field in order to take their similarity into account. When considering the multi-view representation, angular neighbors are especially important for maintaining segmentation consistency among the resulting images. In a graph-cut segmentation approach, objectness cor-

responds to data term and neighbor similarities correspond to smoothness term.

### 3.3. 4D light field segmentation using graph-cut

We formulate the light field segmentation problem as an energy minimization. The energy function consists of two kinds of terms, data and smoothness, and is defined as

$$E(\mathbf{L}) = \sum_{p \in \mathbf{P}} R_p(l_p) + \sum_{\{p,q\} \in \mathbf{N}} B_{p,q} \cdot (1 - \delta(l_p, l_q)), \quad (1)$$

where $E(\mathbf{L})$ is a total energy for the label assignment on a light field $\mathbf{L}$, $l_p$ is the assigned label to the ray at $p$, $\mathbf{P}$ is a set of rays in the light field, $\mathbf{N}$ is a set of neighboring rays, $R_p(l_p)$ is the objectness of the label at $p$, $B_{p,q}$ is the similarity between the rays at $p$ and $q$, and

$$\delta(l_p, l_q) = \begin{cases} 1, & \text{if } l_p = l_q \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

A detailed definition of $R_p(l_p)$ and $B_{p,q}$ will be provided in the following section.

To solve this energy minimization problem, we consider a graph structure with vertices of rays and terminals for each label $\mathcal{O}_i$. There are also two edge types; one connects each ray $p$ and each terminal $\mathcal{O}_i$ with a weight $R_p(\mathcal{O}_i)$, while the other connects neighboring rays $\{p, q\}$ with a weight $B_{p,q}$. Since a light field is irregular, neighboring rays are defined for both spatial and angular. Spatial neighbors correspond to the adjacent pixels in the same viewpoint in a multi-view representation, while angular neighbors correspond pixels in the adjacent viewpoint depending on its disparity. Figure 4 shows the graph structure in a corresponding multi-view representation. In this figure, magenta and orange lines represent spatial and angular neighbors, respectively.

Using the graph structure, we can apply a graph-cut algorithm to minimize the energy function. Eventually, the solution provides the optimal label assignment for each ray. It is noted that the solution becomes near optimal for multi-label segmentation ($n > 2$) [6].

### 3.4. Objectness from intensity and disparity

The energy $R_p(\mathcal{O}_i)$ in Eq. (1) depends on the likelihood (objectness) that the ray $p$ will have the label $\mathcal{O}_i$. As discussed in Sec. 2, it is important to evaluate similarities between the ray and seeds in terms of disparity as well as intensity. Machine learning techniques play an important role in robustly measuring objectness because they can facilitate the integration of these inherently different information types.

To assist data term evaluation, the objectness of the labels should be comparable with each other. We need to choose a machine learning technique that will satisfy this

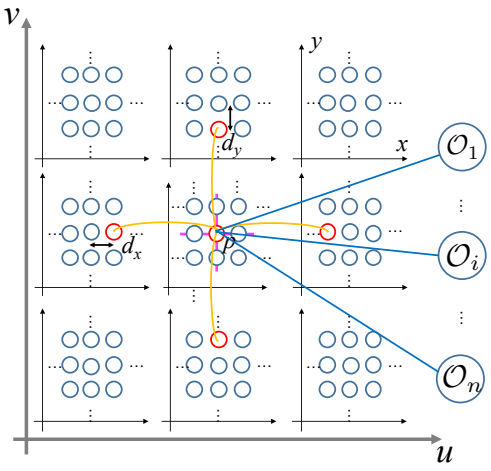



Figure 4: Graph for 4D light field segmentation. The edges are drawn by three different colors. For improved legibility, most of the edges are omitted in this figure and only edges from a single ray $p$ are shown. The blue edges represent the data term. Magenta edges represent spatial neighbor rays. Orange edges represent angular neighbors, which can be determined from the disparity.

requirement while providing good performance. SVM approach is one of popular methods that satisfy this requirement. SVMs maximize the margin between classes hence they generally achieve higher performance, and there are many theoretical studies about SVMs. As for the use of multiple classes in SVMs, comparability between classes is discussed by Platt [24]. Accordingly, we utilize SVMs to define the objectness of the labels. For each objectness of $\mathcal{O}_i$, we train an SVM to treat the seeds of $\mathcal{O}_i$ as positive samples and the seeds of the others $\mathcal{O}_j (j \neq i)$ as negative samples.

Let $\mathbf{x}_p$ be a feature vector of $p$ derived from $I_p$ and $d_p$. By training an SVM using the $\mathbf{x}_p$ of all seeds, we can obtain a weight matrix $\mathbf{w}_i$ and a bias $b_i$ for each label $\mathcal{O}_i$, which represent a decision hyperplane. For a given ray $p$, the SVM calculates the distance between $\mathbf{x}_p$ and the hyperplane, and outputs a decision whether it is positive or negative. Instead of binary decision, we formulate the likelihood as [24]

$$P(\mathcal{O}_i | \mathbf{x}_p) = \frac{1}{1 + \exp(-a f_i(\mathbf{x}_p))}, \quad (3)$$

where $a$ is a step of sigmoid function, and $f_i(\mathbf{x}_{p'})$ represents the distance calculated as

$$f_i(\mathbf{x}_p) = \mathbf{w}_i^\top \mathbf{x}_p + b_i. \quad (4)$$

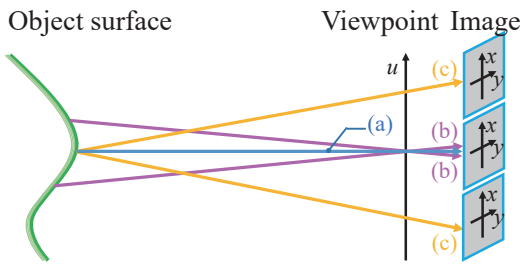



Figure 5: Neighbor relationships in 4D light field. Neighbors of a ray (a) are shown. Rays (b) are spatial neighbors, and rays (c) are angular neighbors.

If $p$ is in a set of seed pixels, we define $P(\mathcal{O}_j|\mathbf{x}_p) = \delta(\mathcal{O}_i, \mathcal{O}_j)$, exceptionally.

As for the feature vector design, we use a joint vector of the chroma and the disparity of the ray $p$ as

$$\mathbf{x}_p = \begin{bmatrix} Cr_p & Cb_p & d_p \end{bmatrix}^\top, \tag{5}$$

where $Cr_p$ and $Cb_p$ are the elements of the YCrCb color space of $I_p$. Chroma is robust to the intensity changes caused by shading effects. Because chroma and disparity have different ranges, it is necessary to normalize them so that the standard deviations become one. We use a Gaussian kernel and a grid search for the normalization.

For translating likelihood to energy, we adopt log-likelihood [4] as

$$R_p(\mathcal{O}_j) = -\ln P(\mathcal{O}_j|\mathbf{x}_p). \tag{6}$$

### 3.5. Smoothness among neighboring rays

In a normal 2D image segmentation problem, smoothness is defined as the color similarity of neighboring pixels. Extending this idea to a 4D light field, light field redundancy should be appropriately considered. As shown in Fig. 5, there are two types of neighboring relationships among the rays. One is *spatial neighbor* as shown as rays (a) and (b), which are the adjacent pixels in the same image in a multi-view representation. Spatial neighbors of $p$ are obtained as

$$q_s = \begin{cases} (u, v, x \pm 1, y) \\ (u, v, x, y \pm 1) \end{cases}. \tag{7}$$

The other one is *angular neighbor* as illustrated as rays (a) and (c), which are rays from the same scene point and observed at adjacent viewpoints. Since all disparities are known, the angular neighbors of $p$ are easily obtained as

$$q_a = \begin{cases} (u \pm 1, v, x \pm [d_p]_X, y) \\ (u, v \pm 1, x, y \pm [d_p]_Y) \end{cases}, \tag{8}$$

where $[d_p]_X$ and $[d_p]_Y$ are $x$ and $y$ components of $d_p$, respectively.

We define the smoothness term for all neighbors $\{p, q\}$ as [4]

$$B_{p,q} = \alpha \exp\left(-g(I_p, I_q)\right), \tag{9}$$

where $g(I_p, I_q)$ is the dissimilarity between $p$ and $q$, and $\alpha$ is a weight with respect to the data term. In our implementation, we evaluate the dissimilarity as

$$g(I_p, I_q) = \frac{(Cr_p - Cr_q)^2 + (Cb_p - Cb_q)^2}{2\sigma^2}, \tag{10}$$

where $\sigma$ controls the rigor of color similarity.

Since the data and smoothness terms evaluate different aspects, a weight $\alpha$ is commonly used to balance them. The dissimilarities between the spatial and angular neighbors can also be used to evaluate different aspects. We control the smoothness weight as

$$\alpha = \begin{cases} \alpha_s & \text{if } q \text{ is a spatial neighbor of } p \\ \alpha_a & \text{if } q \text{ is an angular neighbor of } p \end{cases} \tag{11}$$

## 4. Experiments

We evaluated our method using a public synthetic light field dataset [27], which provides light fields with $9 \times 9$ viewpoints. Four of them (Papillon, Buddha, StillLife, and Horses) have brush stroked seeds for several labels at the center viewpoint image, and the ground-truth labels on the whole 4D light field. In our experiment, we use $5 \times 5$ viewpoints from the $9 \times 9$ light field in order to reduce the data size. The accuracy levels of the obtained results are numerically evaluated by comparing them with the ground-truth.

The disparities are estimated using GCDL [26], which is one of the most well-known disparity estimation method. It is implemented in *cocolib* [10] hence it is easy to use. As previously mentioned, any disparity estimation methods [7, 18, 26] can be used. From these methods, we exploit GCDL for fairness because compared method [28] is build upon GCDL.

We will now discuss experiments, the quantitative comparisons of which are summarized in Table 1. First, we compare the accuracy of our method with that of [28], hereafter referred to GCMLA. GCMLA is also implemented in *cocolib* and we use it for evaluation. The input light field and seeds are shown in the top part of Fig. 7. Since GCMLA only creates for the center image, we compare our center image results with those obtained by GCMLA and found them comparable, as has been summarized in Table 1 and Fig. 7. It is notable that our segmented results for the entire 4D light field also achieved high levels of accuracy.

Next, we evaluate the objectness using SVMs. For comparison purposes, we implemented an alternative objectness using a simple color likelihood [25]. The results, portion of

(a) (b) (c)

Figure 6: Segmentation results of the center image of light field. (a) Our method yields accurate result. (b) Using conventional color likelihood. Some objects are not well separated. (c) Without smoothness constraints, the result become noisy. These results show both learning-based objectness and two smoothness consistencies are effective.

which are shown in Table 1 and Figs. 6a and 6b, indicate that the accuracy obtained using our SVM objectness is significantly higher than the usual color likelihood for all center image and 4D light field data. We can say that by considering disparity differences, our objectness is capable of effectively assigning different labels, even when they have similar color distributions.

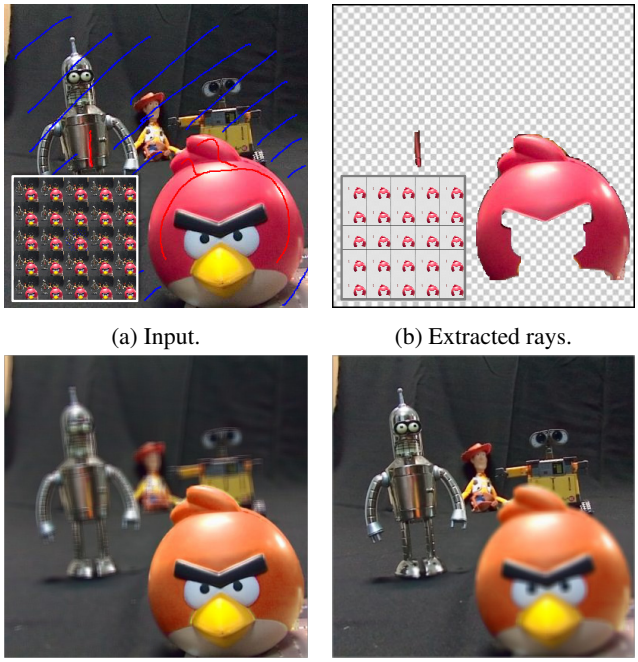
Finally, we evaluate the effectiveness of the use of two smoothness constraints. To accomplish this we compared two types of segmentation results (with and without smoothness constraints). Portion of the results are shown in Table 1 and Figs. 6a and 6c. The results obtained using smoothness constraints are more accurate for both the center image and the 4D light field.

**Application** Our method can be applied to other light field editing applications, such as changing the color of the specific region. After selecting interested region by our method, the color of the region can be easily manipulated in the same manner as image editing.

We applied our method to a light field dataset provided by Jarabo *et al*. [14]. , which are taken under real environments using a Lytro® camera. Figure 8 shows some experimental results of light field editing. Here, we consider a situation that a user wants to manipulate the color of a toy bird. All the user has to do is to give seeds to a portion of the ray as shown in Fig. 8a. After the input, our method extracts rays from the entire light field as shown in Fig. 8b. In this example, we can obtain an edited light field by indicating the hue value. Figures 8c and 8d show refocused images obtained from the edited light field.

## 5. Conclusion

In this paper, we propose a supervised 4D light field segmentation method that can be used for light field editing. Based on user input seeds, objectness of each ray is evaluated using SVMs. To retain redundancy of 4D light fields,



(a) Input. (b) Extracted rays.

(c) Refocus to foreground. (d) Refocus to background.

Figure 8: Light field editing application. (a) input: $5 \times 5$ light field, foreground seed (drawn in red), and background seed (drawn in blue). (b) Foreground ays extracted by our method. (c, d) Refocused images after color manipulation. The toy's color is changed semi-automatically.

we define two neighbors, spatial and angular, and evaluate similarities. By constructing a 4D structured graph, 4D light field can be segmented by a graph-cut algorithm.

The experimental results show that our method achieves higher accuracy than previous methods using public light field datasets. Moreover, we applied our method to real-world light fields and show examples of light field editing. These results show the efficiency of our method for light field editing applications. Light field matting methods [8, 9] can be useful to improve editing quality by applying between segmentation and light field composite pipeline, because they reduce boundary artifacts.

Since we did not apply our method to a wide variety of scenes, one of the future tasks will be validating its effectiveness by applying it to a wider variety of settings, especially non-Lambertian surfaces. Another problem is computational time. Since the use of a graph-cut algorithm requires significant amounts of computational time when numerous vertices are present, an obvious future goal is to solve this problem.

## Acknowledgment

This work was partly supported by JSPS KAKENHI

| Target | 4D light field | | | | In central view-point | | | |
|---|---|---|---|---|---|---|---|---|
| method | GCMLA | color likelihood | w/o smooth | **Ours** | GCMLA | color likelihood | w/o smooth | **Ours** |
| Papillon | - | 79.7 | 94.0 | **98.3** | 97.5 | 79.8 | 94.7 | **98.3** |
| Buddha | - | 85.6 | 94.2 | **97.6** | 96.4 | 85.7 | 94.3 | **97.7** |
| StillLife | - | 94.1 | 95.3 | **96.6** | **96.5** | 94.2 | 95.4 | 96.4 |
| Horses | - | 88.4 | 87.6 | **95.7** | 95.1 | 82.4 | 88.4 | **95.9** |

Table 1: Quantitative comparison with GCMLA [28], color likelihood objectness, without smoothness consistencies, and proposed method. The left part is the segmentation result for the entire light field. Because GCMLA creates only the center image, the segmentation accuracy of the entire light field is not provided. The right part is the segmentation result for the center image. It shows that our method provides higher accuracy than the other method.

Grant Number 25240027, 26700013 from MEXT, Japan.

## References


[1] E. Adelson and J. Bergen. *The plenoptic function and the elements of early vision*, pages 3–20. MIT Press, 1991.

[2] J. Berent and P. L. Dragotti. Unsupervised extraction of coherent regions for image based rendering. In *Proc. BMVC*, 2007.

[3] 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 (TPAMI)*, 34(5):972–986, 2012.

[4] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary region segmentation of objects in N-D images. In *Proc. ICCV*, 2001.

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(9):1124–1137, 2004.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001.

[7] C. Chen, H. Lin, Z. Yu, S. B. Kang, and J. Yu. Light field stereo matching using bilateral statistics of surface cameras. In *Proc. CVPR*, 2014.

[8] D. Cho, S. Kim, and Y.-W. Tai. Consistent matting for light field images. In *Proc. ECCV*, 2014.

[9] J. Fiss, B. Curless, and R. Szeliski. Light field layer matting. In *Proc. CVPR*, 2015.

[10] B. Goldluecke. Cocolib. `http://cocolib.net`.

[11] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The Lumigraph. In *Proc. SIGGRAPH*, 1996.

[12] D. Greig, B. Porteous, and H. Seheult, A. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, (B):271—-279, 1989.

[13] D. R. Horn and B. Chen. LightShop: interactive light field manipulation and rendering. In *Proc. Symposium on Interactive 3D Graphics and Games*, 2007.

[14] A. Jarabo, B. Masia, A. Bousseau, F. Pellacini, and D. Gutierrez. How do people edit light fields? *ACM Transactions on Graphics (ToG)*, 33(4):1–10, 2014.

[15] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proc. ECCV*, 2002.

[16] A. Kowdle, S. N. Sinha, and R. Szeliski. Multiple view object cosegmentation using appearance and stereo cues. In *Proc. ECCV*, 2012.

[17] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. SIGGRAPH*, 1996.

[18] H. Lin, C. Chen, S. B. Kang, and J. Yu. Depth Recovery from Light Field Using Focal Stack Symmetry. In *Proc. ICCV*, 2015.

[19] K. Maeno, H. Nagahara, A. Shimada, and R. Taniguchi. Light Field Distortion Feature for Transparent Object Recognition. In *Proc. CVPR*, 2013.

[20] K. Marx. The Focused Plenoptic Camera. In *Proc. ICCP*, 2009.

[21] T. Nagahashi, H. Fujiyoshi, and T. Kanade. Spatio-temporal volume Video Segmentation Using Iterated Graph Cuts Based on Spatio-Temporal Volume. In *Proc. ACCV*, 2009.

[22] R. Ng, M. Levoy, G. Duval, M. Horowitz, and P. Hanrahan. Light Field Photography with a Hand-held Plenoptic Camera. In *Technical Report CSTR*, 2005.

[23] S. Osher and J. a. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[24] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[25] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut" - Interactive foreground extraction using iterated graph cuts. In *Proc. SIGGRAPH*, 2004.

[26] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4D light fields. In *Proc. CVPR*, 2012.

[27] S. Wanner, S. Meister, and B. Goldluecke. Datasets and Benchmarks for Densely Sampled 4D Light Fields. In *Proc. Vision, Modelling and Visualization*, 2013.

[28] S. Wanner, C. Straehle, and B. Goldluecke. Globally consistent multi-label assignment on the ray space of 4D light fields. In *Proc. CVPR*, 2013.

[29] Y. Xu, H. Nagahara, A. Shimada, and R. Taniguchi. TransCut: Transparent Object Segmentation from a Light-Field Image. In *Proc. ICCV*, 2015.

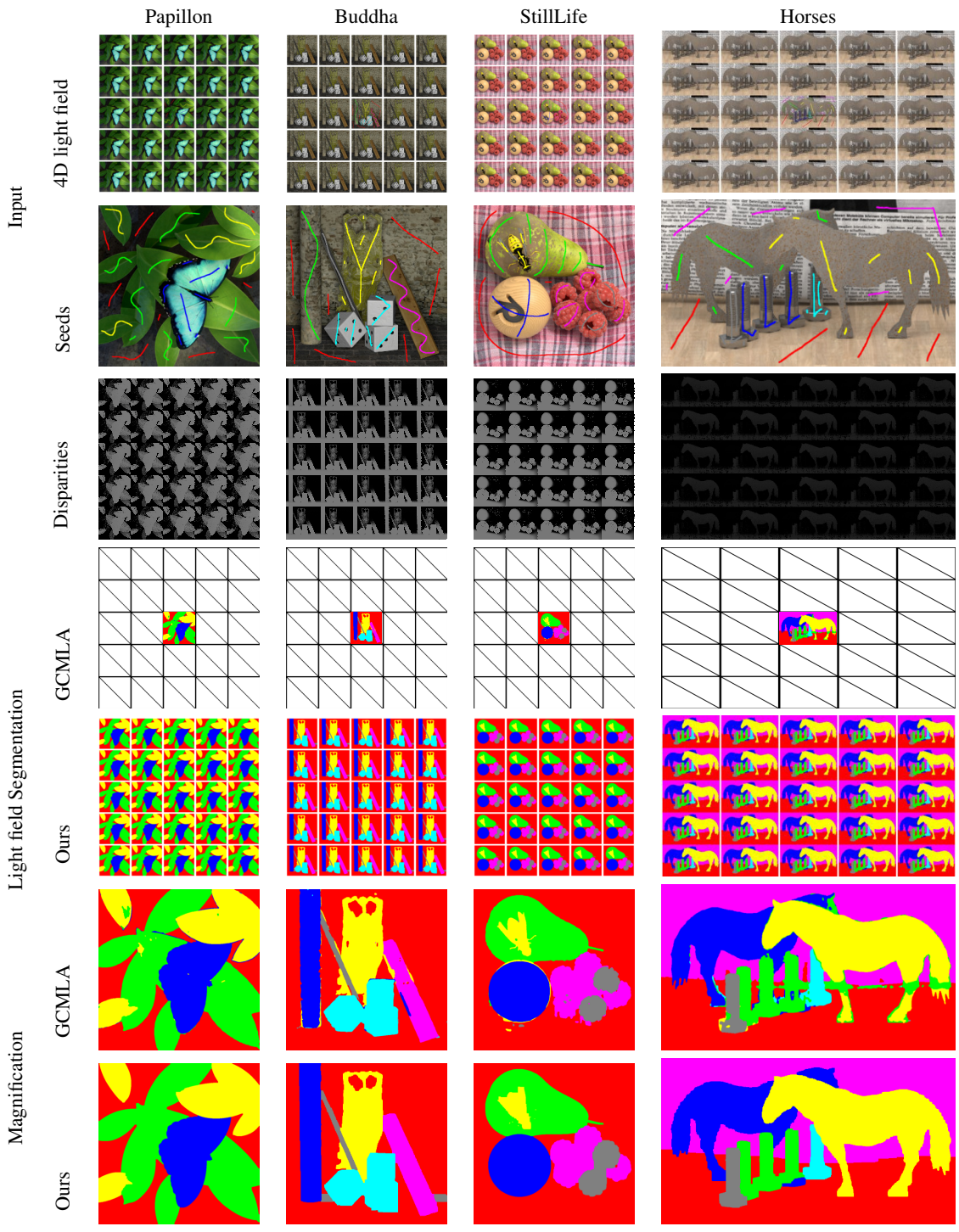



Figure 7: Input and segmentation results. The 1st row shows the input $5 \times 5$ light field. The 2nd row shows the user input seed, which is given in the center image of the input. The 3rd row shows the estimated disparity by GCDL [26]. The 4th and 5th rows show 4D segmentation results. The 6th and 7th rows show center image magnifications of the 4th and 5th rows, respectively.