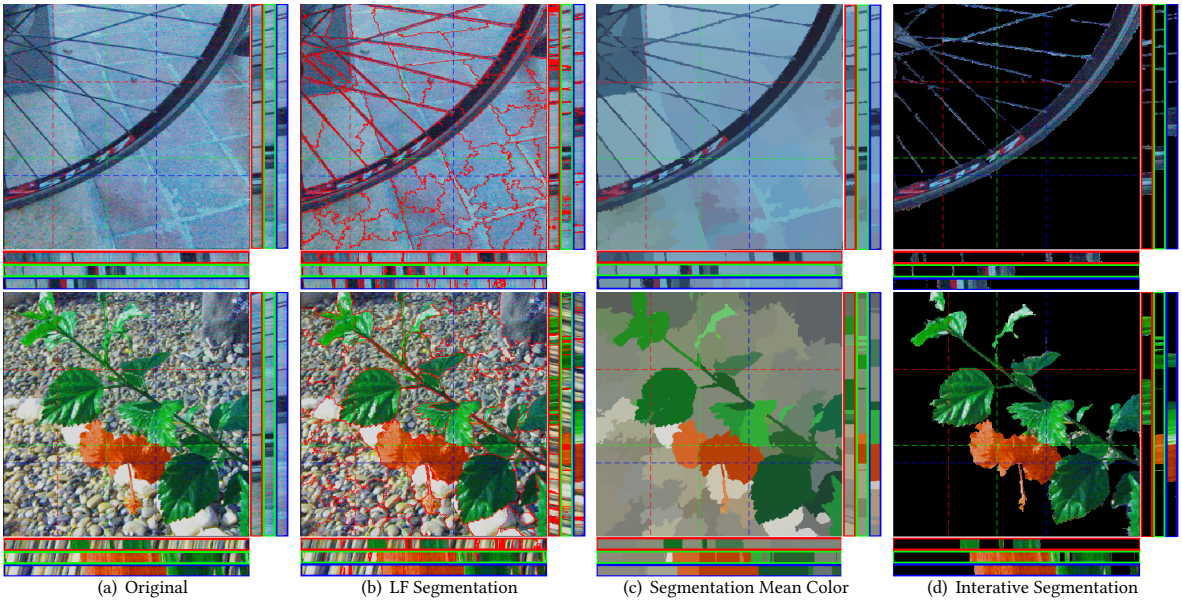
Hierarchical and View-invariant Light Field Segmentation by Maximizing Entropy Rate on 4D Ray Graphs

RUI LI and WOLFGANG HEIDRICH, King Abdullah University of Science And Technology, Saudi Arabia



(a) Original (b) LF Segmentation (c) Segmentation Mean Color (d) Interative Segmentation

Fig. 1. Our light field segmentation method facilities segmentation of fine structures in light fields with complex occlusions and difficult textures. Here we show from left to right: two source light fields, the achieved segmentation, mean color regions, and object selection (which requires additional user input). We also show the EPIs for different horizontal and vertical slices.

Image segmentation is an important first step of many image processing, computer graphics, and computer vision pipelines. Unfortunately, it remains difficult to automatically and robustly segment cluttered scenes, or scenes in which multiple objects have similar color and texture. In these scenarios, light fields offer much richer cues that can be used efficiently to drastically improve the quality and robustness of segmentations.

In this paper we introduce a new light field segmentation method that respects texture appearance, depth consistency, as well as occlusion, and creates well-shaped segments that are robust under view point changes. Furthermore, our segmentation is hierarchical, i.e. with a single optimization, a whole hierarchy of segmentations with different numbers of regions is available. All this is achieved with a submodular objective function that allows for efficient greedy optimization. Finally, we introduce a new tree-array type data structure, i.e. a disjoint tree, to efficiently perform submodular optimization on very large graphs. This approach is of interest beyond our specific application of light field segmentation.

We demonstrate the efficacy of our method on a number of synthetic and real data sets, and show how the obtained segmentations can be used for applications in image processing and graphics.

CCS Concepts: • **Computing methodologies** → **Image processing**; *Computational photography*.

Additional Key Words and Phrases: Light Fields, Segmentation

ACM Reference Format:
Rui Li and Wolfgang Heidrich. 2019. Hierarchical and View-invariant Light Field Segmentation by Maximizing Entropy Rate on 4D Ray Graphs. *ACM Trans. Graph.* 38, 6, Article 167 (November 2019), 15 pages. https://doi.org/10.1145/3355089.3356521

Authors' address: Rui Li, rui.li@kaust.edu.sa; Wolfgang Heidrich, wolfgang.heidrich@kaust.edu.sa, King Abdullah University of Science And Technology, Thuwal, 23955-6900, Saudi Arabia.

0730-0301/2019/11-ART167 $15.00
https://doi.org/10.1145/3355089.3356521

1 INTRODUCTION

Segmentation is a canonical problem in visual computing, and a key component of many techniques used in computer graphics, image processing, and computer vision (Fig. 1). Many previous works address segmentation problems by splitting 2D image regions based on low-level similarity [Achanta et al. 2012; Felzenszwalb and Huttenlocher 2004; Liu et al. 2011], mid-level structure [Arbelaez et al.

2011; Khan et al. 2015, 2017], or high-level semantic meaning [Long et al. 2015]. State-of-the-art superpixel segmentation methods can achieve strong results on most general cases. Despite this progress, there are still many difficult corner cases that will lead to segmentation failure, e.g., when an object has a similar appearance to the background, when occlusion occurs, or when an object has a complex shape. Under these challenging situations, the usual object priors (appearance similarity, contrast, regular shape) are insufficient, and 2D image segmentation can fail as a result. 3D scene information can be used as a powerful evidence to further remove segmentation ambiguity. However, 2D image segmentation approaches naturally lack 3D scene information, and the inference of 3D information from 2D images is challenging, inconvenient, and expensive.

In recent years, 4D light fields [Levoy and Hanrahan 1996] have become more popular for many tasks, as they not only encode spatial information but also parallax, and can be recorded with special cameras [Ng et al. 2005; Venkataraman et al. 2013; Wilburn et al. 2005]. Examples for uses of light fields to solve problems in graphics and vision include synthetic aperture imaging [Isaksen et al. 2000], segmentation [Hog et al. 2016], saliency detection [Li et al. 2014], multi-object detection [Pei et al. 2012], and visual odometry [Dansereau et al. 2011]. Wu et al. [2017] provides a recent and detailed review for light fields.

Many graphics applications can potentially benefit from light field segmentation. 3D reconstruction from multiple frames [Yücer et al. 2016] requires a light field segmentation as input. 4D segmentation can be considered as a low level clustering method to facilitate light field video compression [Miandji et al. 2019]. For tomographic applications, 4D segmentation can be a benefit for multi-frame super resolution [Zang et al. 2018a], moreover, dynamic reconstruction [Zang et al. 2018b, 2019] could reduce computational primitive to accelerate its speed or refine reconstruction details by separating data into small full 4D segmentation.

The parallax information in light fields implicitly encodes 3D scene information as well as object scene position, making it highly suitable for more robustly solving challenging segmentation cases, e.g., occlusion, foreground-background similarity, and so forth. However, light fields not only provide new cues, but they also bring new challenges in terms of computation time and storage cost. Due to redundant image view information, loading and processing light field data requires large computational and memory resources.

In this paper, we explore the way to utilize redundant angular view information and higher dimensional representations in the light field image domain, and proposed a submodular objective function to solve the 4D light field segmentation problem. A small sample of our results is shown in Fig. 1.

Specifically, our contributions are as follows:

- We expand traditional color and texture appearance terms to also account for depth consistency and align region boundaries with occlusion edges.
- We introduce a new prior on Epipolar Plane Images that encourages view consistency of the segmentation, i.e. that the same object point remains assigned to the same region as the viewpoint changes.
- The segmentation method is formulated as the maximization of a graph entropy on an undirected weighted graph in the 4D light field domain. We formulate the segmentation problem as splitting a graph into several subgraphs to obtain a higher entropy rate, where each subgraph is a segmentation. This produces a *segmentation hierarchy* which avoids experimentation with different segmentation parameters.
- Since this is a submodular problem, a greedy heuristic optimization scheme can guarantee a bound of $(\frac{1}{2})$ on estimating a globally optimal solution [Liu et al. 2011, 2014], with a computational complexity of $O(N \log N)$ and a memory complexity of $O(N)$.

2 RELATED WORK

In this section, we give brief reviews for 2D, video, and 4D light field segmentation.

Image Segmentation. In general, image segmentation utilizes object priors to split image pixels into several non-overlapping regions. There are several main classes of methods to solve this problem. We only provide a terse overview and refer the interested reader to David et al. [2018] for more comparisons of 2D segmentation methods. Graph-based methods construct a graph to describe the similarity of neighboring pixels [Felzenszwalb and Huttenlocher 2004] or to find a uniform disjoint pixel set [Liu et al. 2011]. Paris and Durand [2007] use Morse theory to interpret the mean shift as a topological decomposition into density modes, which then generates hierarchy of regions. Edge-based methods utilize an edge detector as a cue for where to place region boundaries [Arbelaez et al. 2011]. The assumption behind edge-based methods is that objects have strong edges surrounding them. However, this assumption sometimes leads to over-segmentation and tends to fail when the object blends into the background. Region-based methods try to find segmentations with similar statistical properties, e.g., color or other statistical attributes [Van den Bergh et al. 2012], [Chen et al. 2017]. SLIC [Achanta et al. 2012] is one of the most famous superpixel methods, which adopts k-means clustering using color and spatial position as features, and they adopt the mean of feature space to describe the region property. To liberate the user from tuning superpixel size and number, Achanta et al. [2018] proposed a method to automatically adapt to the local texture and scale of an image.

Video Segmentation. Video segmentation also has a long tradition in the research community. Due to the additional temporal dimension, video segmentation has additional cues, such as motion, disparity, frame-coherence, which can be applied to handle more complex cases. For example, Grundmann et al. [2010] proposed an efficient and scalable method by first over-segmenting a volumetric video graph into spatial and temporal 3D superpixels, and then iteratively merging these. Ayvaci and Soatto [2012] described a video segmentation method that separates surfaces in the scene that are partially surrounded by integrating appearance and motion into the objective function. Chen et al. [2017] simultaneously predicts pixel-wise object segmentation and optical flow in videos that based on a fully convolutional network and a FlowNet model.

Wang et al. [2015b] solves video segmentation with a saliency approach by considering two discriminative visual features: spatial edges and temporal motion boundaries as indicators of foreground object locations.

Light Field Segmentation. While video segmentation is able to utilize extra information over 2D image segmentation, as discussed above, this information is not very structured – videos can exhibit complex patterns of different object and camera motions. On the other hand, 4D light fields provide parallax from multiple views for a stationary scene. The highly structured parallax information also makes it easier to analyze angular structures spanning multiple views. Xu et al. [2015] proposed a transparent object segmentation by utilizing consistency and distortion on 4D light fields. Yücer et al. [2016] present a 3D reconstruction algorithm to automatically segment a static foreground object from a highly cluttered background using a hand-held camera. This method uses the coherence of data in the light field to reveal extra structure. Hao et al. [2017] proposed a very efficient 4D light field superpixel method that considers the invariance to refocusing. However, their method requires extra depth information as input. It also still has difficulties when background and foreground objects share similar textures, or when the reflectance of an object is non-Lambertian. Hog et al. [2016] proposed a novel graph representation for interactive light field segmentation with a human in the loop. The graph structure exploits redundancy in the ray space in order to reduce computational primitives. Later, Hog et al. [2017] proposed an automatic light field segmentation by clustering super-rays and light ray bundles. The clustering metric relies on spatial, angular and color distance, to align multiple view points. Extra sparse view disparity estimation is utilized for more accurate ray alignment. Wanner et al. [2013] also proposed a ray based segmentation method to solve the multi-label segmentation problem using a variational framework. Mihara et al. [2016] proposed a learning-based light ray detection method by utilizing appearance and disparity cues, and then adopted a graph-cut framework to solve segmentation.

Our core method can be seen as an extension of the 2D image segmentation of Liu et al [2011] to 4D light fields, with additional energy terms that utilize geometric structure in the higher dimensional space. While our method takes an initial depth estimate as an input (in our implementation, this depth estimate is generated by the method of Tao et al. [2013]), our light field segmentation model does not make a strong assumptions on light ray constancy or EPI constancy, which does not hold when occlusion occurs. As we show in Section 6, this approach results in significantly improved view consistency, as well as better alignment of the segmentation regions with object boundaries compared state of the art methods. Finally, we also significantly improve compute time and memory consumption of submodular segmentation frameworks like Liu et al [2011], as well as introduce a hierarchical version of this framework that minimizes iterative parameter adjustments.

3 PRELIMINARIES

In the following we briefly summarize relevant concepts for our work and introduce notation used throughout the paper.

3.1 Light Fields

Throughout this work, we denote 4D light fields as $L(x, y, u, v)$, where (u, v) can be interpreted as the coordinates of a view point and (x, y) as the 2D image coordinates on a focus plane located at unit distance from the view point plane (see Fig. 2).

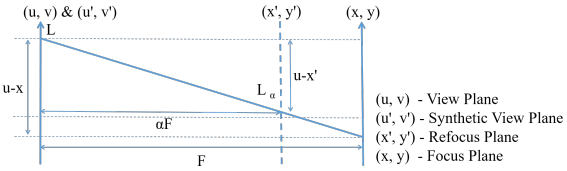


Fig. 2. Diagram of light field geometry and notation used throughout the paper. The (u, v) plane contains the view points, while the (x, y) plane at unit distance is the canonical focus plane of the light field. The light field can be refocused to a different plane (x', y') at distance α according to Eqn. 1.

Light Field Refocusing. In a refocusing operation, the light field is re-parameterized for a different location α of the (x, y) plane. This re-parameterization step can be expressed as a counter-clockwise shear of the Epipolar Plane Images (EPIs) [Ng et al. 2005; Wang et al. 2015a]:

$$L_\alpha(x', y', u, v) = L(u + \frac{x' - u}{\alpha}, v + \frac{y' - v}{\alpha}, u, v). \quad (1)$$

View plane (u, v) and synthetic view plane (u', v') are actually co-located in our setting, since only the refocus plane moves while (u', v') is fixed in Fig. 2. We use same coordinates system as Ng et al. [2005] for light field refocusing.

Light Field Slicing. For our segmentation method, we analyze 2D slices of clusters that emerge when fixing two of the light field parameters. We denote these slices of the light field domain as, $L^{x,y}(u, v), L^{u,v}(x, y), L^{x,u}(y, v)$, and $L^{y,v}(x, u)$. Note that $L^{u,v}(x, y)$ corresponds to a perspective image, while $L^{x,u}(y, v)$ and $L^{y,v}(x, u)$ respectively denote the horizontal and vertical EPIs.

3.2 Light Field Graphs and Submodular Functions

Graph Structure. We denote a graph on a light field as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i | i = 1, \ldots, N\}$ is the vertex set, which is composed of a regularly sampled grid of points v in the 4D ray space, i.e. each node in the graph corresponds to one ray. The edges $e \in \mathcal{E}$ connect the immediate neighbors along all four dimensions of the light field and also include a self-loop connecting each ray to itself. Also see Fig. 3(left).

The edge weight represents an affinity between vertices, and is a function $w : \mathcal{E} \rightarrow \mathbb{R}^+ \cup \{0\}$. Moreover, a disjoint division of the vertex set $\mathcal{V}$ forms a graph partition $\mathcal{S} = \{s_1, s_2, \ldots, s_k\}$, where i is the partition index. Our goal is to select a subset of edges $\mathcal{A} \subseteq \mathcal{E}$ such that the resulting graph $(\mathcal{V}, \mathcal{A})$ consists of K connected components (i.e., K regions), i.e. $\mathcal{A}$ is missing all edges that cross regions.

In analogy to light field slicing operators defined above, we also define slices of the clusters, $s_k^{x,y}, s_k^{u,v}, s_k^{x,u}, s_k^{y,v}$.

Submodular functions. A submodular function is a set function $\mathcal{F} : 2^V \rightarrow \mathbb{R}$ that has diminishing returns property, i.e.,

$$\mathcal{F}(A \cup \{a_1\}) - \mathcal{F}(A) \geq \mathcal{F}(A \cup \{a_1, a_2\}) - \mathcal{F}(A \cup \{a_2\}), \quad (2)$$

where A is a set and $a_1, a_2 \notin A$. This property can be utilized as an efficient way to greedily optimize the objective function by finding the element with maximal energy gain. This guarantees a $(1 - \frac{1}{e})$-approximation of the global optimum [Krause and Golovin 2014].

Entropy rate of a random walk on a weighted graph. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that has N vertices $\{v_i | i = 1, \ldots, N\}$ and edge weights $w_{i,j} \geq 0$. A random walk starting from an initial vertex to other vertices in the graph can be described by a sequence of vertices $\{X_t | t \in \{1, \ldots, N\}\}$. Given the current position v_i, the next position v_j is chosen according to the weight of edges that connected to the vertex v_i, with transition probabilities $p_{i,j} = \frac{w_{i,j}}{\sum_k w_{i,k}}$. The stationary distribution $\mu P = \mu$ is given by

$$\mu = (\mu_1, \ldots, \mu_N)^T = (\frac{w_1}{\overline{w}}, \ldots, \frac{w_N}{\overline{w}})^T, \quad (3)$$

where $w_i = \sum_k w_{i,k}$ and $\overline{w} = \sum_i w_i$ [Cover and Thomas 2006].

The entropy rate of a random walk on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be written as a set function:

$$H(\mathcal{E}) = -\sum_i \mu_i \sum_j p_{i,j} \log(p_{i,j}), \quad (4)$$

where log refers to the logarithm with base 2.

Similar to Liu et al. [2011], the transition probability of the segmented graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ is defined as:

$$p_{i,j} = \begin{cases} \frac{w_{i,j}}{w_i}, & \text{if} \quad i \neq j \text{ and } e_{i,j} \in \mathcal{A}, \\ 0, & \text{if} \quad i \neq j \text{ and } e_{i,j} \notin \mathcal{A}, \\ 1 - \frac{\sum_{k:e_{i,k} \in \mathcal{A}} w_{i,k}}{w_i} & \text{if} \quad \text{i = j.} \end{cases} \quad (5)$$

4 LIGHT FIELD SEGMENTATION MODEL

Like 2D image segmentation methods, light field segmentation should produce clusters with consistent colors and textures. The clusters should be well shaped, and of similar size. In addition, however, we can define several additional desired properties that are specific to light field segmentation and can be used to obtain superior results:

(1) Depth-Awareness: should be able to separate objects with similar appearance according to scene depth.
(2) Occlusion-Awareness: clusters should have sharp edges at occlusion boundaries.
(3) View Consistency: the clusters should be stable and consistent under changes in view points.

Our segmentation method *maximizes* an objective function of the form

$$E(\mathcal{A}) = A(\mathcal{A}) + \lambda V(\mathcal{A}) + \beta C(\mathcal{A}) + \gamma S(\mathcal{A}), \quad (6)$$

where $A(\mathcal{A})$ is an occlusion and depth aware appearance term, $V(\mathcal{A})$ encourages view consistency, $C(\mathcal{A})$ regularizes the spatial shape, and $S(\mathcal{A})$ encourages similarly sized clusters.

The four terms are defined in detail in the following subsections. Each term is monotonic and submodular, and therefore $E(\mathcal{A})$ can be maximized by an efficient greedy scheme.

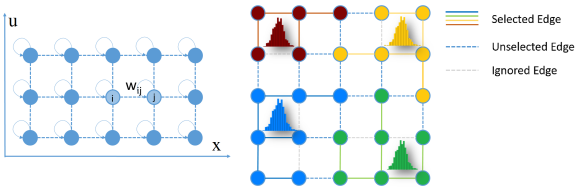


Fig. 3. An illustration of our two-stage appearance model. **Left**: at the first stage, we construct a graph on a regular grid of 4D ray space, where the appearance weight for edges is simply determined by the color of individual rays. **Right**: when the total number of clusters is reduced beyond a threshold (e.g., 2 times of spatial resolution), the appearance weight is jointly determined by ray intensity and each clusters' intensity histogram. Moreover, we re-initialize all the unselected edge weights when activating the second stage.

4.1 Occlusion- and Depth-aware Appearance Term

In light field segmentation, refocused depth can be a good indicator to handle strong texture and isolated objects that are placed at different scene depth. Moreover, occlusion boundaries can also preserve sharp edges for segmentation.

Our edge weight function combines the above information, and is defined as

$$w_{i,j} = \frac{w^a_{i,j} + w^d_{i,j}}{1 + k w^o_{i,j}}, \quad (7)$$

where i and j represent neighboring rays in the 4D ray space, w^a is an appearance weight, w^d a depth weight, and w^o an occlusion weight. These three individual weights are defined below. Fig. 3 illustrates our graph structure over the ray space in flat land. In 4D, the nodes are connected along all 4 dimensions, x, y, u, and v. According to Eqn. 7, pixel and local histogram features determine edge weight calculation which shows on the right of Fig. 3.

With these edge weights we can define the transition probabilities $p_{i,j}$ as in Eqn. 5 and the graph entropy $H(\mathcal{A})$ as in Eqn. 4. The first term of our objective function is given as

$$A(\mathcal{A}) = H(\mathcal{A}). \quad (8)$$

Since the entropy rate of a random walk on the graph is a monotonically increasing submodular function [Liu et al. 2011, 2014], the inclusion of any unselected edge will lead to an increase of the entropy on the graph. However, this increase is lower when selecting edges from the set of remaining edges due to the diminishing return property.

Appearance Weight. The appearance weight w^a measures the similarity in appearance between two rays using an appearance function $D(.)$:

$$w^a_{i,j} = \exp(-\frac{D(i,j)}{2\sigma^2}), \quad (9)$$

with

$$D(i,j) = \|f(i) - f(j)\|_2^2 + \eta \|hist(s(i)) - hist(s(j))\|_2^2, \quad (10)$$

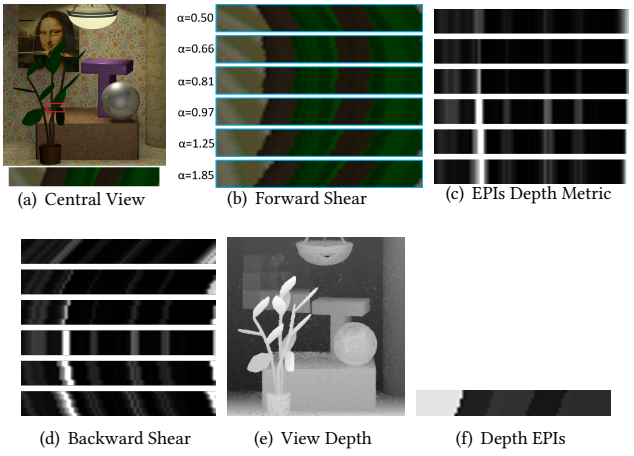
Here, the first term simply compares the colors of rays i and j. The second term is a texture feature in the form of a color histogram for the two clusters $s(i)$ and $s(j)$. The parameter η weights these two distance terms. We use two different values for η: for small clusters,

the number of rays in the cluster is insufficient for obtaining robust histogram statistics, so we set $\eta = 0$. For larger clusters (more than 2 wide in both x and y) we set $\eta > 0$ to enable region and texture descriptors. This term uses *Lab* space for color and histogram comparisons.

Depth Weight. The depth weight w^d encourages that rays which intersect the scene at similar depths are placed into the same cluster:

$$w^d_{i,j} = \exp(-\frac{\|d(i) - d(j)\|_2^2}{2\sigma^2}). \tag{11}$$

Here, $d(i)$ is a depth estimate for each ray i, which we obtain as follows. We first shear the light field to refocus it at different candidate depths. We then use refocused depth cues from Tao et al. [2013] to determine for each (x, y) location the candidate depth that brings it into the best focus. Finally, we propagate this information back to 4D by backward shearing the result onto the original light field (see Fig. 4).



(a) Central View (b) Forward Shear (c) EPIs Depth Metric

(d) Backward Shear (e) View Depth (f) Depth EPIs

Fig. 4. Computing the 4D depth cost by forward and backward shearing. 4(a) shows a central view and EPIs. We first apply forward shearing (i.e., refocusing) on the original light field 4(b) with a set of candidate depths, and calculate the light field depth metric for each 4D ray element 4(c). We then apply backward shearing of the 4D depth metric 4(d) to the original light field parameterization. Finally, the 4D refocused depth can be estimated by finding the minimum of the per-element depth metric from different shearing parameters. 4(e) and 4(f) shows the central view of 4D refocused depth and EPIs.

Occlusion Weight. Finally, our graph weights consider depth discontinuities as well as intensity edges. Specifically, we define

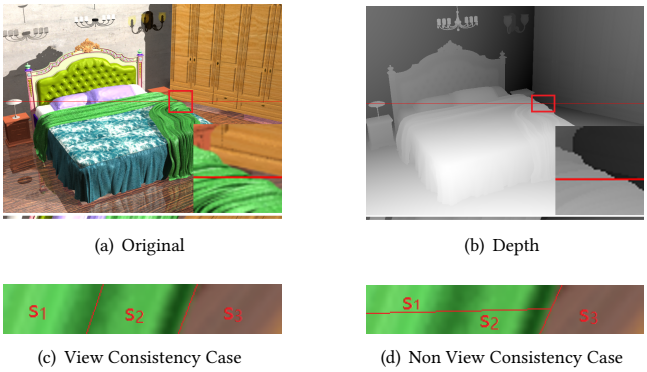
$$w^o_{i,j} = |o_d(i) \cdot o_L(i) - o_d(j) \cdot o_L(j)|. \tag{12}$$

Here, $o_L(.)$ is simply an edge detector [Dollár and Zitnick 2015] applied to each (x, y) slice, while o_d is computed based on 4D gradient magnitude of depth estimates from above:

$$o_d = \begin{cases} 1 & ; \frac{||\nabla d||}{d} \geq 0.8, \\ 0 & ; \text{else} \end{cases} \tag{13}$$

The depth threshold here is quite conservative, since it it intended mainly to detect occlusion events.

4.2 View Consistency Term



(a) Original (b) Depth

(c) View Consistency Case (d) Non View Consistency Case

Fig. 5. An illustration of view consistency. (a) and (b) are the example cases of RGB synthetic scene and depth scene. (c) and (d) show two possible segmentations in light field EPIs slices. (c) preserves a good View Consistency compared to (d). (d) has a disconnected light ray; the same object point is assigned to different clusters in different views. This is discouraged by our view consistency term.

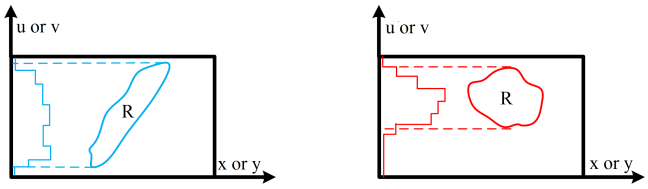
A light field segmentation should be view consistent, i.e. under gradual changes in viewpoint, the 2D slice of the segmentation in the projected image should not change abruptly; the same object points should be grouped into the same cluster under all views. This means that the segmentation should be encouraged to cut along the spatial (x, y) dimensions instead of the (u, v) dimensions for clusters with similar appearance. Fig. 5 illustrates this principle: assuming that regions s_2 and s_3 have similar appearance and scene depth, we prefer the left segmentation from Fig. 5(c) over the one from Fig. 5(d).

Fig. 6 shows an example comparison of the view consistency term. The "thin and tall" case better preserves view consistency and has a higher numeric values than "fat and short" case, which leads to a higher priority to be selected in submodular optimization. To measure the view consistency of a 4D region, we propose a new metric based on the entropy rate of 2D (x, y) slices for a fixed view point, $s_k^{u,v}(x, y)$. The proposed view consistency constraint favors segmentations where these slices are of uniform size and shape as the view point changes. Specifically, we define the distribution of 2D segmentation slice as follows,

$$p_{s_k}(u, v) = \frac{|s_k^{u,v}|}{|s_k|}, \tag{14}$$

where $|\cdot|$ represents elements number. Then, view consistency term is defined as the entropy rate of $p_{s_k}(u, v)$, i.e.

$$V(\mathcal{A}) = -\sum_k \mu_{s_k} \sum_{u,v} p_{s_k}(u, v) \log p_{s_k}(u, v). \tag{15}$$

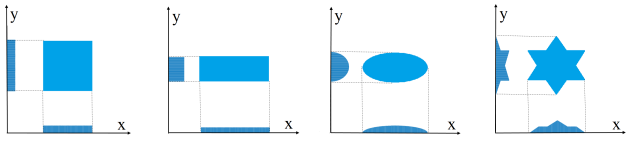


(a) VC Constraint = 1.536 (b) VC Constraint = 0.719

Fig. 6. An illustration of the Light Field view-consistency constraint. Fig. 6(a) and Fig. 6(b) shows two different light field segmentations with similar size, 4D pixels will be projected to 2D angular coordinates and count the occurrence. (a) ("thin and tall") has a more uniform angular distribution compared to (b) ("fat and short"), thus, (a) has a higher photo-consistency objective value.

4.3 Spatial Compactness Term

As we have just seen, elongated cluster shapes along the u and v directions are desired in the EPI. However, the cluster shape in the spatial x, y slices is preferred to be round and compact without complicated boundaries. This is in agreement with the goals of traditional 2D image segmentation methods. Again we measure the spatial shape regularization by the entropy rate of spatial distribution of clusters. We first project our cluster elements onto the x and y axes, and then measure the entropy rate of the cluster in terms of x and y coordinates. To provide an intuition of our spatial shape regularization, we illustrate the entropy rate of several shapes in Fig. 7. To measure the entropy rate of spatial distribution, we count



(a) C=15.34 (b) C=14.85 (c) C=14.71 (d) C=15.17

Fig. 7. Illustration of the light field spatial compactness term. We project a region onto the (x,y)-coordinates and enhance spatial compactness by maximizing the entropy rate of the spatial shape distribution. 7(a) shows that the square shape of segmentation has a higher objective than the other shapes, and is thus preferred by our system.

coordinate histogram of x and y by fixing remaining coordinates as

$$p_{s_k}(y,u,v) = \frac{|s_k^{y,u,v}|}{|s_k|}, \quad p_{s_k}(x,u,v) = \frac{|s_k^{x,u,v}|}{|s_k|}, \tag{16}$$

where $p_{s_k}(y,u,v)$ and $p_{s_k}(x,u,v)$ are s_k's slices along the x and y coordinates respectively. The shape regularization terms for the two axes are then

$$C_x(\mathcal{A}) = -\sum_k \mu_{s_k} \sum_{y,u,v} p_{s_k}(y,u,v) \log p_{s_k}(y,u,v), \tag{17}$$

$$C_y(\mathcal{A}) = -\sum_k \mu_{s_k} \sum_{x,u,v} p_{s_k}(x,u,v) \log p_{s_k}(x,u,v). \tag{18}$$

The overall shape regularization is the sum of these two terms:

$$C(\mathcal{A}) = C_x(\mathcal{A}) + C_x(\mathcal{A}). \tag{19}$$

4.4 Size Balancing Term

Finally, we follow [Liu et al. 2011] and regularize cluster size as an additional constraint. The entropy rate of the balancing term is given by

$$S(\mathcal{A}) = -\sum_k \mu_k \log(\mu_k) - |\mathcal{S}|. \tag{20}$$

When maximized, this term encourages equally sized clusters. The stationary distribution of 2D slice is

$$\mu_{s_k} = \frac{|s_k|}{|\mathcal{V}|}. \tag{21}$$

5 LARGE-SCALE HIERARCHICAL SUBMODULAR OPTIMIZATION

Performing standard submodular optimization on a light field graph will lead to a large-scale submodular optimization problem. Currently available light field data will generate graphs with up to 10^9 edges, while 2D images only have 10^7 edges due to the more densely connected graph structure in 4D vs 2D. We develop several strategies to deal with large-scale optimization problems involving submodular functions. These include a disjoint tree as a hierarchical way of re-generating a specific number of regions, as well as a cache system to efficiently search, modify, merge large-scale trees. These innovations will be useful for other large-scale submodular optimization problems including for videos and volumes.

Hierarchical Segmentation with the Disjoint Tree. One challenge for segmentation problems is always to guess the correct number of regions. One solution is to generate a hierarchical segmentation, and let user adjust the number of regions by selecting the appropriate level in the hierarchy. We design the disjoint tree to record the tree merging procedure. The advantage of the disjoint tree is that it does not require a large amount of memory to store a segmentation hierarchy, but can recover any number of segmentations by simply providing the sequence of merged trees without having to recompute the segmentations. The disjoint tree is a variation of a standard disjoint set, which has a tree-like structure that preserves information about the order in which elements are added. A disjoint tree is a set of binary trees. Merging two trees requires connecting the root node of one tree to the empty child node of another tree, and saving the root id of the merged tree. Submodular optimization involves selecting the optimal trees to merge until there is only one tree left. Given the final tree and the merged root node order, we can search and decompose the tree node reversely, so that any number of trees (regions) can be recovered. This is illustrated in Fig. 8.

Max Heap and Partial Update Scheme. For each iteration of Alg. 1, only the edge with maximal energy gain is selected and the energy gain of the rest of edges will be updated. Naive implementation will cost $O(|\mathcal{E}|)$ iterations to find an optimal edge, and $O(|\mathcal{E}|)$ iterations to update the rest of the edges, i.e. the computational complexity will be $O(|\mathcal{E}|^2)$. This is too expensive for light fields, which have tens of millions of pixels. Fortunately, submodular functions

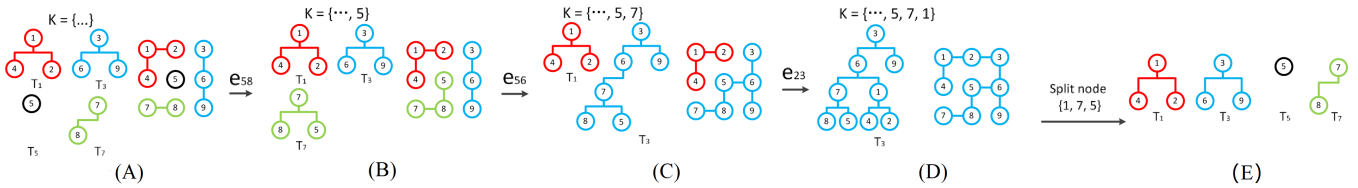


Fig. 8. An illustration of merging in the disjoint tree and recovering hierarchy of segmentation by providing merged region sequences. T_i is one binary tree of the disjoint tree with root node id i. When an optimal edge is selected (e_{58}, e_{56}, e_{23}) for each iteration, two connected binary trees (T_5, T_7), (T_3, T_7), (T_1, T_3) will be merged respectively. The merged binary tree forms a sequence $\{T_5, T_7, T_1\}$, i.e., $\mathcal{K} = \{\ldots, 5, 7, 1\}$. The hierarchy of segmentation regions can be recovered by disconnecting the disjoint tree in reverse order. For example, if we disconnect tree nodes 1 from its parent, the disjoint tree in (D) will roll back to (C), and disconnect nodes 7 and its parent, disjoint tree in (C) will roll back to (B) and so on. Any possible number of regions can be recovered given the final disjoint tree and the sequence of merges.

have a diminishing return property; the energy gain for each edge will never be increased during the iteration. This property enables a lazy, greedy method [Leskovec et al. 2007; Liu et al. 2011] for quick search and update. The basic idea is to adopt a max heap structure to determine the optimal edge by popping the top element of the heap, and only partially recalculating the elements near the top of the heap until the max heap requirements are again satisfied. However, a naive implementation of max heap may lead to a worst case of $O(|\mathcal{E}|^2 \log |\mathcal{E}|)$ when the binary max heap becomes unbalanced. The updating max heap will cost up to $O(\log |\mathcal{E}|)$, depending on the tree depth. In a small dataset, max heap will only update several times in general, and therefore $\mathcal{E}$ is generally negligible. However, in large-scale data such as in light fields, max heap updates may happen many times for each iteration. We make improvements in two ways: First, the max heap will be re-initialized several times via fully updating all tree nodes and re-balancing to avoid the worst case. Therefore, the actual update times will be limited strictly. Second, our method will drop all redundant edges (edges that connect the same subgraph) in the disjoint-tree. Therefore, the size of max heap will reduce dramatically and will decrease accordingly.

Efficient Tree Operation Using Caching. Our submodular function is optimized with a disjoint tree and a max heap, therefore, the optimization process involves tree node search, add, modify, find operations. For large tree sizes, the naive implementation of tree operations will lead to system stack overflow due to the highly recursive nature of the method. In our implementation, tree operations are recursion free and apply Breadth-First-Search or Depth-First-Search to implement the various tree operation. Moreover, we design a cache system to search for empty child nodes: instead of searching the tree from the root node, candidate empty child nodes can be directly popped from a cache queue.

Memory-Efficient Greedy Heuristic Optimization. The aim of our proposed optimization scheme is to maximize our submodular function and preserve the hierarchical structure of the segmentation. The submodularity of the objective function leads to a good guarantee for estimating the global optimum by greedy optimization [Liu et al. 2011]. The algorithm starts with an empty set (i.e., $\mathcal{A} = \emptyset$), each vertex is totally disconnected, and then iteratively finds the largest energy gain edge. An edge that forms a cycle on the edge set $\mathcal{A}$ is not ignored immediately. We maintain disjoint pixels set by disjoint tree, if there is a selected edge to be added to the edge set $\mathcal{A}$, this procedure will lead to the merging of two binary trees (no cycle constraint). We also record the order of the binary tree merging process $\mathcal{K} = \{K_t | t = 1, \ldots, |\mathcal{V}|\}$. The iteration converges when there is only one binary tree in the disjoint tree. The desired number of regions K is obtained by splitting nodes in the disjoint tree in reverse order to form new binary trees. The pseudo code is shown in Alg. 1. Optimizing entropy rates on graphs with submodular functions is a standard optimization tool. Our work improves the traditional pipeline by jointly generating hierarchy and objective update scheme, caching system for large-scale tree operation, which makes large-scale tree-based submodular optimization to be memory tractable and computationally efficient.

Data: 4D light field $L(x, y, u, v)$
Result: Disjoint Tree $\mathcal{T}$, order set $\mathcal{K}$
Initialize:
$\mathcal{T} = \{\mathcal{T}_i | i = 1, \ldots, |\mathcal{V}|\}$,
$\mathcal{A} = \emptyset$, $\mathcal{U} \leftarrow \mathcal{E}$, t = 0, $\mathcal{K} = \emptyset$.
while $\mathcal{U} \neq \emptyset$ **do**
  $e^* = \arg\max_{e \in \mathcal{U}} E(\mathcal{A} \cup \{e\}) - E(\mathcal{A})$.
  if e^* *connects* $\mathcal{T}_i$ *and* $\mathcal{T}_j$ $(i \neq j)$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{e^*\}$
    $\mathcal{T}_i \leftarrow \mathcal{T}_i + \mathcal{T}_j$ (+ stands for tree merge operator)
    $\mathcal{K}_{t+1} = j$, $t = t + 1$
  end
  $\mathcal{U} \leftarrow \mathcal{U} - \{e^*\}$
end

Algorithm 1: 4D Light Field Segmentation Algorithm

6 EXPERIMENTS

In this section, we show the visual results and quantitative comparison for 4D segmentations of both real and synthetic light fields. Please refer to supplemental material for additional results and comparisons.

Datasets. In the experiment, we mainly use several publicly available datasets: the **4D Light Field Dataset** [Honauer et al. 2016], the **CVPG Dataset** [Zhu et al. 2017], as well as the **Stanford Light Field Archive** [Dansereau et al. 2019]. For the **4D Light Field Dataset**, we select data with a segmentation mask for evaluation. Of these, the first two mainly provide synthetic datasets with highly

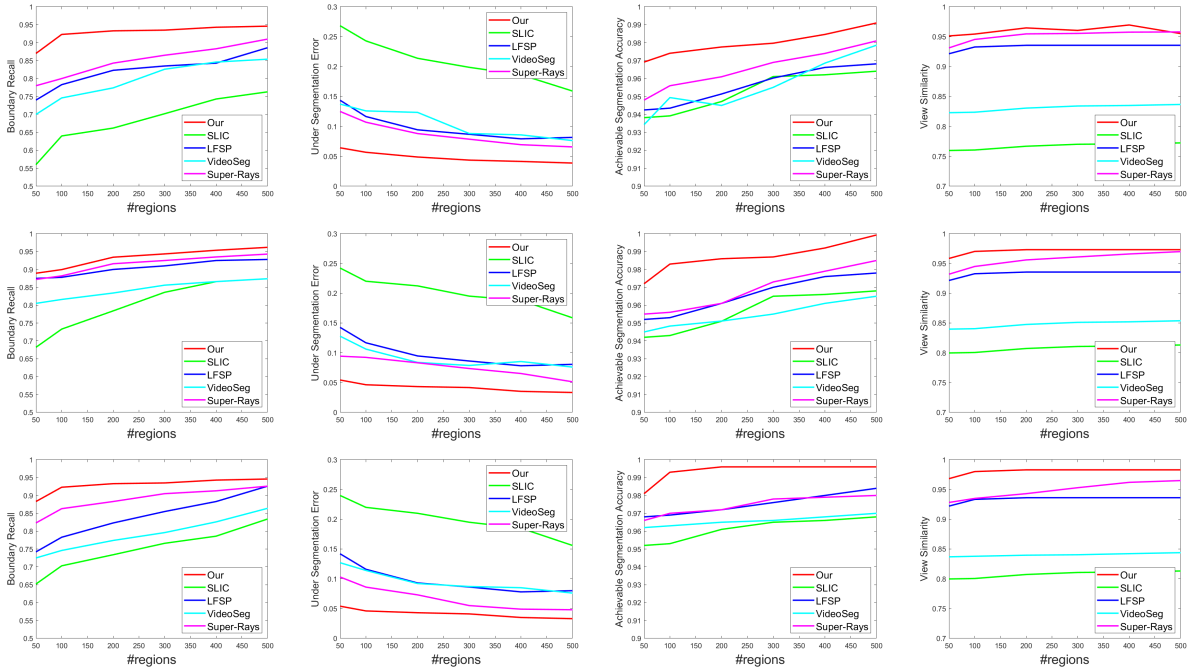


Fig. 9. Quantitative comparison of our method with four competing methods: SLIC [Achanta et al. 2012], LFSP [Zhu et al. 2017], Super-Rays [Hog et al. 2017] and VideoSeg [Grundmann et al. 2010]. The different comparison metrics are shown from left to right: Boundary Recall, Under-Segmentation Error, Achievable Segmentation Accuracy, and our View Similarity metric. Each graph shows the progression of each metric as the number of regions in the segmentation is increased. Different datasets are shown from top to bottom: CVPG dataset, 4D Light Field Dataset, and our own dataset.

accurate ground truth for disparity and segmentations. The **Stanford Light Field Archive** contains examples of very high angular resolution light fields, where it is a challenge for segmentation methods to preserve view consistency. Also, system issues such as memory efficiency are highlighted by this dataset. The **CVPG Light Field Dataset** [Zhu et al. 2017] provides 6 synthetic light field datasets with ground truth disparity and 4D segmentation labels. We augment these standard datasets with our own dataset, captured using a first generation Lytro camera. This dataset includes a manually labeled ground truth 4D segmentation and will be made public.

Parameter Settings. In the experiments, we set weight parameters in our objective functions weight as $\lambda = 0.1$, $\beta = 0.1$, $\gamma = [0.5, 100]$, $\sigma = 0.005$, $k = 0.5$, $\eta = 0.5$.

6.1 Quantitative Evaluation

Evaluation Metric. To evaluate segmentation performance, we employ widely used 2D segmentation evaluation metrics: boundary recall (BR), under-segmentation error (UE), and achievable segmentation accuracy (ASA), which are all standard metrics in 2D image segmentation [Achanta et al. 2012; Liu et al. 2014; Zhu et al. 2017]. To evaluate the view consistency of the light field segmentation for different views, we design a new View Similarity (VS) metric as follows. Given a segmentation, we determine for each region s_k and each pair of views (u, v) and (u', v'), the region slices $s_k^{u,v}$ and $s_k^{u',v'}$. These are then aligned in image space for maximum pixel overlap, before the metric is computed as

$$VS = \frac{1}{N_{vs}} \sum_k \sum_{u'} \sum_{v'} \sum_u \sum_v \frac{|s_k^{u',v'} \cap s_k^{u,v}|}{|s_k^{u',v'} \cup s_k^{u,v}|}, \tag{22}$$

where $\cap$ and $\cup$ are the union and intersection of the pixel regions *after finding the best alignment between the slices by 2D translation in image space*, $N_{vs} = |\mathcal{S}| \cdot N_A \cdot N_A$, and N_A is the number of views in the light field.

Quantitative results. Figure 9 shows the quantitative comparison of our method on the three datasets that have ground truth information. The comparison methods are VideoSeg [Grundmann et al. 2010], a state-of the art video segmentation method, and finally LFSP [Zhu et al. 2017] and Super-Rays [Hog et al. 2017], two recent automatic light field segmentation methods. Note that we do not compare against other light field segmentation methods that require manual user input. More detail on the chosen comparison methods is provided in the next section. Of the four metrics, lower values are better for UE while for the other three metrics (BR, ASA, and VS), higher values are better. The results clearly demonstrate that our method outperforms the comparison methods on all metrics. Compared to image and video segmentation methods, the light field methods (LFSP, Super-Rays, and ours) are able to use the richer information of light fields for better segmentation results. At the same time, our submodular energy term proves more effective than

the competing Light field approaches. Please refer to the supplement for additional quantitative results on other datasets. We use the implementation of the metrics from the segmentation toolkit [Stutz et al. 2018].

Due to the hierarchical nature of our light field segmentation we only need to run the method once on each light field and then can extract the segmentations with different numbers of regions to produce the data in Fig. 9. All other methods need to be run many times with adjusted parameters to produce the same data. A single execution takes on average 4 min for LFSP, 2 min for SLIC, and 20 min for VideoSeg, and 1 min for Super-Rays. More details of computational time comparison is shown in Table 1. This compares to 5 min for the full hierarchical segmentation in our approach.

Trade-off Between Region Number and Shape. The graphs in Fig. 9 show evaluation metrics against the number of regions. Our evaluation covers different possible applications of segmentation: superpixels, smoothing and denoising [Bi et al. 2015], semantic segmentation [Fulkerson et al. 2009] etc. For semantic object segmentation, a small number of regions is preferable ($100 \sim 200$) to capture small objects or components. For smoothing or editing tasks, fine image structure should be preserved, therefore, a large number of regions (> 200) is better in general. Overall, our method is the most competitive algorithm against other state-of-the arts in the major range of region number Fig. 9.

Earlier methods (e.g., SLIC, Videoseg) have a worse performance when only requiring less than 200 regions, because those methods generate very compact regions with simple boundaries, which means that boundary adherence is poor for segmentations with a small number of regions. LFSP and Super-Rays are two state-of-the-art competitors, these two methods share similar performance and visual behavior due to use SLIC-like data term (color and spatial distance) with extra LF-based constraint. Like SLIC, the compact region shape is not suitable for faithfully representing objects in a light field with a small number of regions. The view consistency metric is not significantly affected by the region number, and Superrays, LFSP and our method achieve a better performance than those traditional 2D segmentation methods. Our method shows superior performance against all state-of-the-art methods numerically and visually due to our approach of directly optimizing a full 4D graph.

Computation Time. Our method emphasizes reconstruction quality over speed, and as such cannot match the speed of some existing methods, especially those optimized for GPU computation. Nonetheless, the comparison in Table 1 shows that the execution times our method is within a small factor of the times for other CPU methods. All experiments were conducted on a workstation with an Intel Xeon E5-2687 CPU (3.0GHz) and 192 GB RAM. In evaluating the the computational efficiency of the algorithms, we also note that the all comparison methods produce a single segmentation would in many practical settings need to be re-run multiple times to determine the appropriate number of clusters for a given light field. By comparison, our hierarchical approach produces all possible segmentation granularities in a single run within the time listed in the table, thus significantly reducing the amount of time needed to conduct experiments.

Table 1. Compute time comparison (Angular Resolution is 9×9).

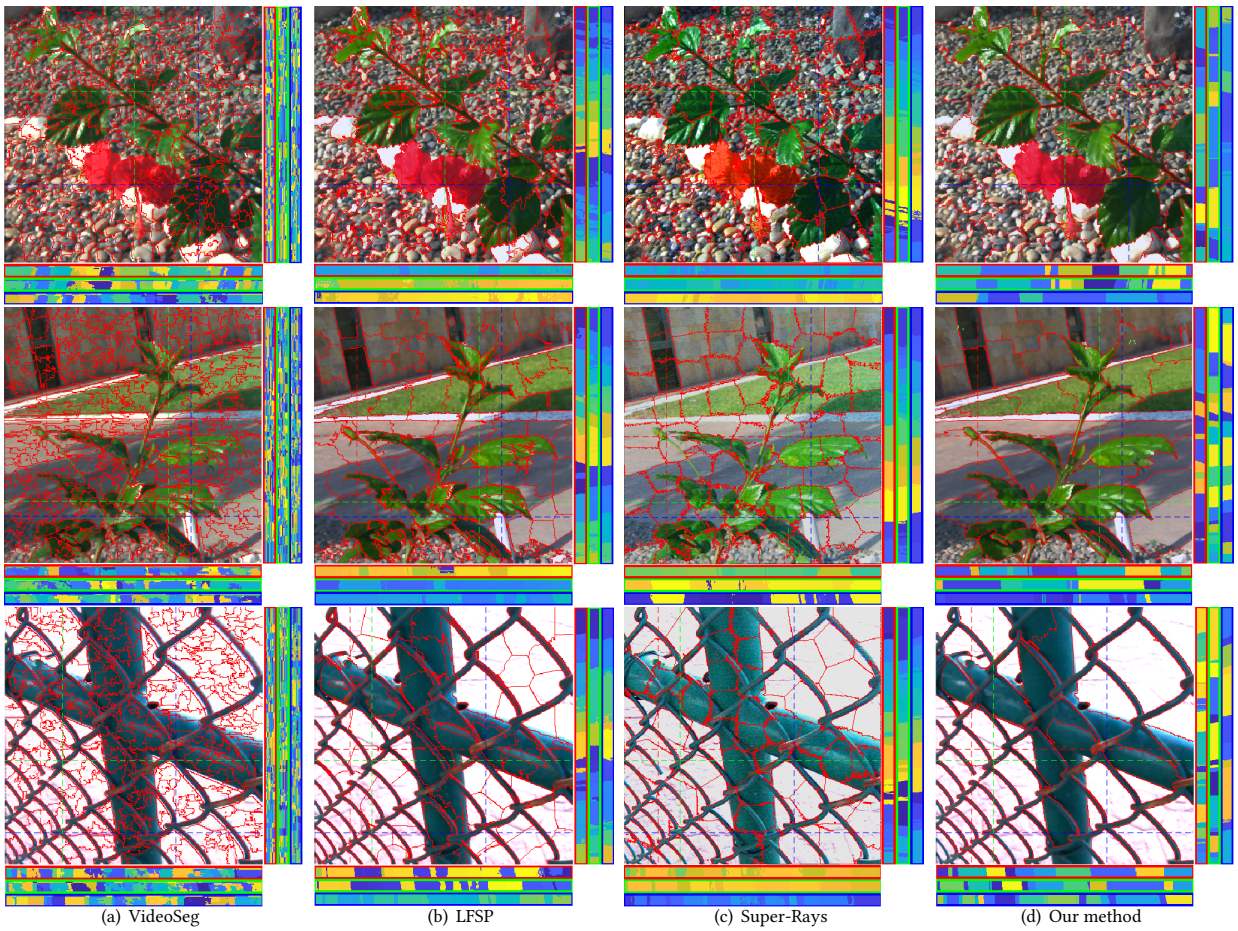
Method	378×378	375×540	512×512
SLIC 4D	102s	124s	180s
Videoseg	~900s	~1020s	-
LFSP	142s	162s	192s
Super-Rays	34s	42s	61s
Our	177s	289s	434s

6.2 Visual Comparison

We visually compare our light field segmentation method with other state-of-the art alternatives. We perform comparisons on three datasets: 4D Light Field Dataset, Stanford Light Field Archive, and our dataset. We strongly encourage the reader to also refer to the video and supplemental material for more dynamic visualizations.

Visual 4D Segmentation Comparison. Fig. 10 and Fig. 11 show visual comparisons with state-of-the-art 2D, video, and light field segmentation methods. We simply rearrange light field to form a sequence of views, and then apply a video segmentation method to process the light field sequence. VideoSeg [Grundmann et al. 2010] is a popular video segmentation method that utilizes appearance similarity and optical flow to group small regions. Video segmentation exploits inter-frame optical flow and appearance similarity to separate objects that contain motion in the video sequence. However, in light field sequence, optical flow on different view points may contain discontinuities, especially when light field sequence change view from right to left or down to up, this may affect the grouping of small regions. Moreover, a light field has comparatively low variance in terms the disparity, therefore, traditional optical flow methods may not have obvious flow output from neighboring view points. LFSP [Zhu et al. 2017] builds a 4D segmentation in a SLIC-like fashion, with added view invariance constraints. Its performance is similar to SLIC in the spatial dimensions. Since LFSP requires a depth map, we first compute depth the same way as in Section 4, and provide it as input to LFSP. LFSP relied rather heavily on this depth map, while our method builds the segmentation from full 4D ray space (not from grouping superpixels). The core difference between the LFSP [Zhu et al. 2017]'s view consistency and ours is that our term can guarantee a good error bound and that it is a dense pixelwise constraint on region shape to preserve view consistency in EPIs and spatial slices, while theirs can only apply a constraint on the segmentation centroid, which is sparse and lacks control over region shape.

Super-Rays [Hog et al. 2017]'s code is not public available, we re-implemented their work. Their pipeline is quite fast, stable and simple. Since super-rays metric is also based on SLIC-like spatial and color distance, this metric tends to generate very regularly shaped spatial regions, at the expense of alignment with boundary features, which becomes particularly apparent when segmenting fine spatial features (also see Fig. 12). As a result, our method is more robust to mistakes in depth map, and achieves better view consistency in 4D domain. By jointly considering appearance, depth, occlusion as well as the prior knowledge of 4D segmentation shape, we obtain



(a) VideoSeg (b) LFSP (c) Super-Rays (d) Our method

Fig. 10. Visual comparison with state-of-the-art methods on our real scene dataset. We visualize 3 different alternative methods: VideoSeg[Grundmann et al. 2010], LFSP[Zhu et al. 2017] and Super-Rays[Hog et al. 2017]. To emphasize the view consistency of segmentation, we also show the EPIs on vertical and horizontal directions (e.g., RED, GREEN, BLUE dash line and EPIs with same color rect). The number of regions are around 100 for all the alternatives.

better robustness in complex scenes and gain the ability to find fine structures in the light field.

Depth Refinement. Segmentation can be utilized as extra cue to refine depth boundary, interior noisy region and depth holes. Fig. 12 illustrates the depth refinement results by state-of-the-art LF segmentation methods and our proposed methods. We simply utilize median depth as output value, and then adopt light field refocused depth estimation [Wang et al. 2015a] as raw depth input. When scenes light field contain small structures (first row) or lens glare (second row), traditional depth metrics will tend to fail on such cases. Fortunately, our light field segmentation was able to utilize view consistency constraints to remove short-term lens flashing and preserve fine structure of tiny scene, e.g., black holes in depth map (second row) can be fixed by our segmentation methods.

Hierarchical Segmentation. The hierarchical aspect of the segmentation allows us to very efficiently adjust the number of regions without having to re-run the segmentation. One straightforward application is simply to minimize times for experimentation. In our experience, the results in many previous image/video/LF works require extensive experiments parameter adjustments to produce good results, and only the time to produce the final result is actually reported. Our approach can overcome this issue. Examples of this adjustment are shown in the video and supplementary materials.

6.3 Ablation Study

We conduct several ablation experiments to test our objective components: ablation study by canceling objective terms, ablation study by varying weights, and quantitative evaluation. We simply canceling objective weights (i.e., λ, β, $\gamma = 0$) to show the actual behavior of individual terms, which shown in Fig. 13(a). The example figure contains smooth region (sky), complex texture (grass) and fine structure. Our full method with all objective components can handle the above cases, and canceling view consistency terms will lead to a large amount of inconsistent small regions Fig. 14, which may

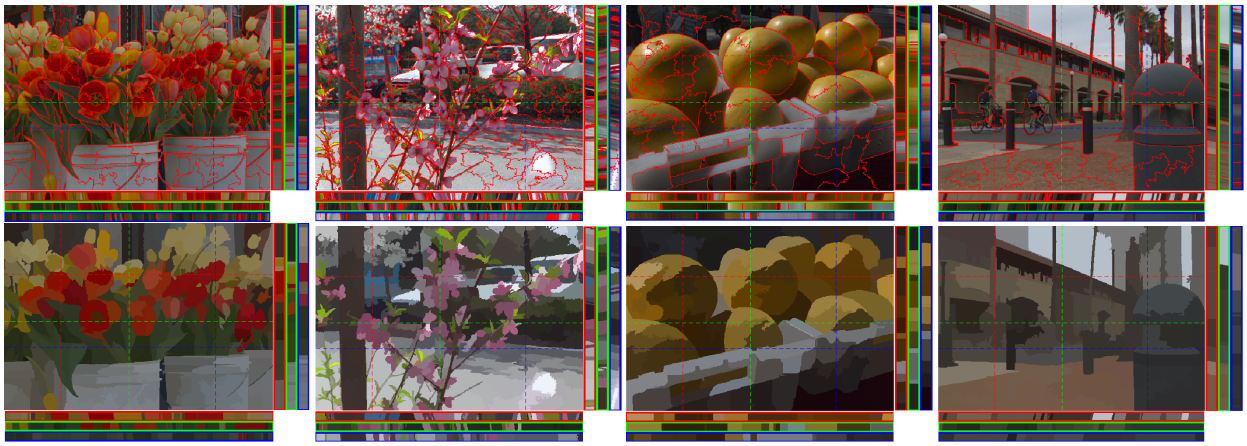
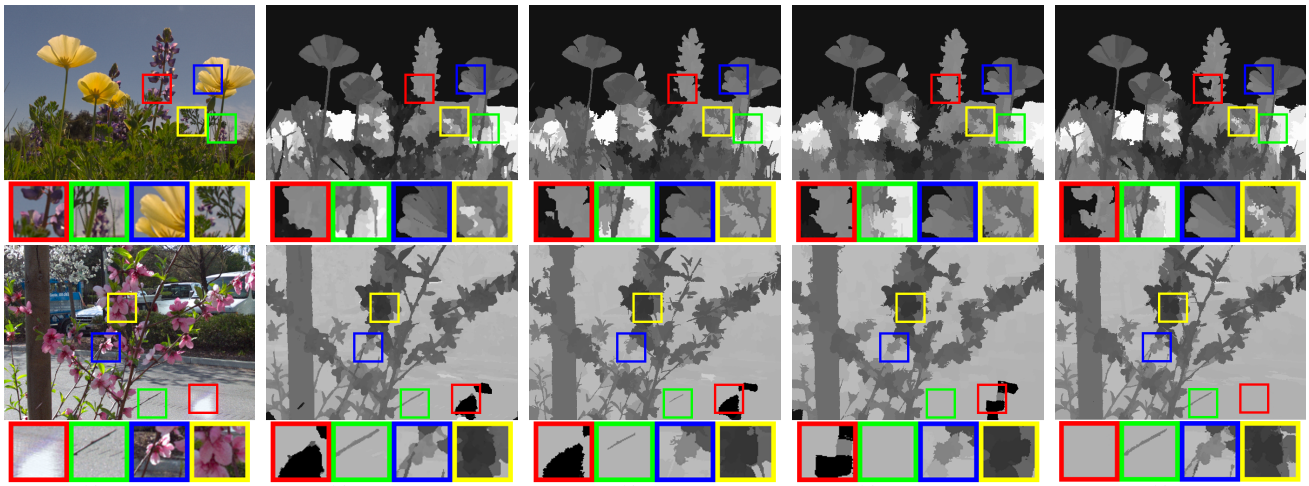


Fig. 11. Additional visual results. From left to right, we show original views, segmentation contour, random color and mean color within segmentations with EPIs.

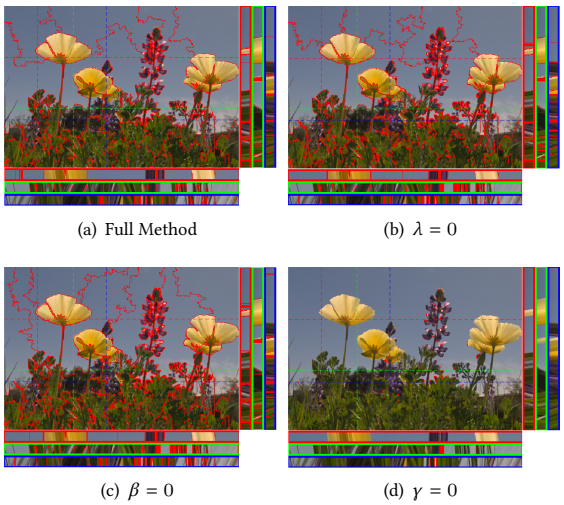


(a) Original (b) Input Depth (c) LFSP (d) Super-Rays (e) our

Fig. 12. Comparison of depth maps encoded in the segmentations created by different methods. (a) Original central scene view. (b) Input raw depth for our method. Depth results for (c) LFSP [Zhu et al. 2017], (d) Super-Rays [Hog et al. 2017], and (e) our method. Our method shows significantly better preservation of thin structures and straight edges in both example scenes, demonstrating a better alignment of segmentation boundaries with scene structure. The red crop regions in both scenes show that our method manages to refine and correct errors in the initial depth map. In the top scene, a region of the sky is erroneously merged with the flower, while in the bottom example, lens flare creates an erroneous floating structure in the original depth estimates. Both of these artifacts are corrected by our segmentation method.

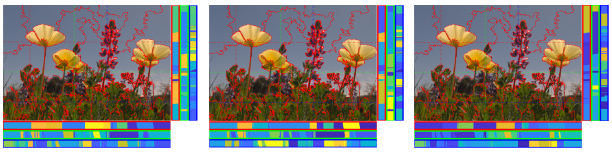
disappear in other view. Fig. 13(c) shows that the removal of spatial compactness term will lead to a irregular shape, especially on the smooth region. Fig. 13(d) illustrates that size balancing term regularize the imbalanced size of region, canceling it will generate tiny small regions.

Fig. 14 varies view consistency weight λ, small inconsistency may appear in some of the views without view consistency constraints, and the increasing of λ significantly removes those small noise regions. Fig. 15 compares different spatial compactness weights β, the increasing of β provides higher strength of spatial regularization to separate smooth regions (e.g., sky), which mainly generates region with concentrating spatial distribution. Fig. 16 shows the results of varying γ to generate different granularity of region. Quantitative results of the ablation study are shown in Fig. 17, showing the different quality metrics for different combinations of parameters λ and β. Overall, a fine-tuned γ can improve BR, UE and ASA, since this term provides extra prior to remove false segmentation between multiple frames. β only regularize shape and generates visually better results (especially in non-boundary or non-texture parts), but does not significantly change evaluation metrics.



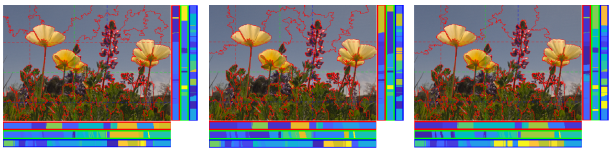
(a) Full Method (b) $\lambda = 0$ (c) $\beta = 0$ (d) $\gamma = 0$

Fig. 13. Ablation Test. (a) example results with all objectives. (b) the removal of view consistency term. (c) the removal of spatial compactness term. (d) the removal of size balancing term.



(a) $\lambda = 0.001$ (b) $\lambda = 0.05$ (c) $\lambda = 0.1$

Fig. 14. Ablation test by varying λ for view consistency test. (a) Grass regions contain small inconsistency due to a low λ. (b) increasing λ for inconsistency removal. (c) higher strength of inconsistency removal.

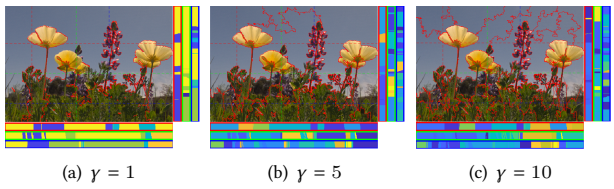


(a) $\beta = 0.001$ (b) $\beta = 0.01$ (c) $\beta = 0.1$

Fig. 15. Ablation test by varying β for spatial shape regularization. (a) segmentation will not be a central symmetry shape with a small β in smooth region (i.e., sky). (b) increasing β. (c) regular and symmetry shape of region in smooth and edgeless regions with a higher β.

6.4 Failure Cases and Limitations

We show failure cases of wide baseline and non-Lambertian in Fig. 18 with color EPIs. Certain types of light field can lead to a failure of our method, e.g., wide baseline light fields captured by captures by camera arrays. In this scenario, the EPI is severely undersampled, so that the tracking of view consistency can fail. Non-Lambertian reflectance causes several challenges, including non-uniform appearance, which affects the initial depth estimation, but also the view consistency term. Our method can find the correct edge of object, however the segmentation within the object can be fail since the appearance is not consistent under view point changes. Another limitation of our method comes from submodular optimization, the weights (i.e., λ, β, γ) for each objective components need to be normalized, therefore, these parameters will be affected by extrema in objective values due to noise or light exposure.



(a) $\gamma = 1$ (b) $\gamma = 5$ (c) $\gamma = 10$

Fig. 16. Ablation test by varying γ for elements number in each regions. (a) small value of γ will have a more flexible segmentation but tends to ignore small region (b) when increasing γ, the region size will tend to be uniform and balanced.

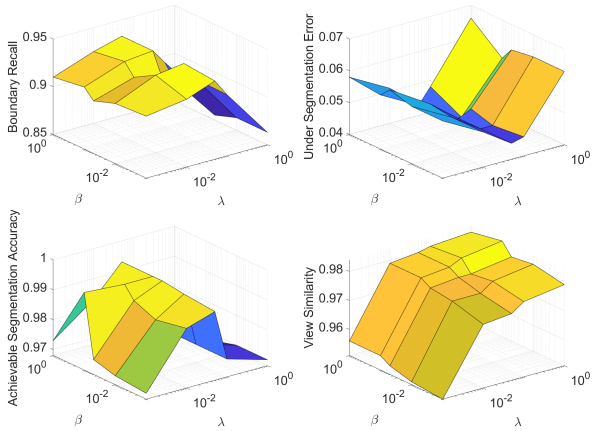


Fig. 17. Ablation test for the quantitative evaluation of varying γ, λ and β. We set parameters pairs as the combination three weight choices γ={0.1, 1, 5, 10, 100}, λ={0.001, 0.01, 0.05, 0.1, 1}, β = {0.001, 0.01, 0.05, 0.1, 1}, region number is 100.

7 APPLICATIONS

Segmentation is a starting point for many processes in image manipulation and computer vision. In the following we highlight several applications of our light field segmentation.

7.1 User-guided Object Segmentation.

Like most 2D and video segmentation methods, our method segments the light field into regions of consistent appearance, but not into semantic objects. However with a simple user interface, we can can manually select multiple regions that comprise a single object. Examples of this user-guided object segmentation are shown in Fig. 19.

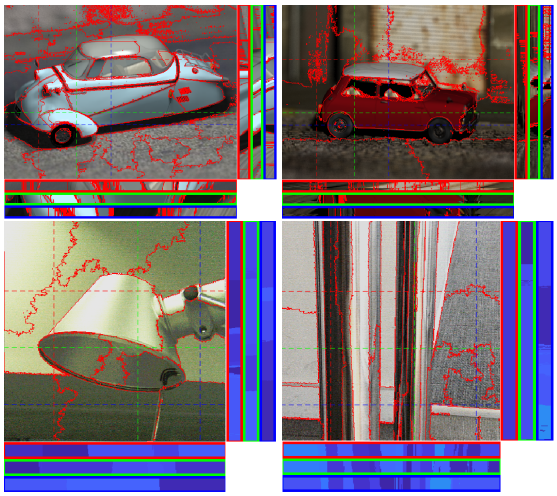


Fig. 18. Failure case illustration. First Row: we test our method in wide baseline cases of light field, segmentation tends to inconsistent due to large view point changes. Second Row: the visual results of non-Lambertian cases. Our method discovers correct object boundaries, but mistakenly separates objects due to incorrect depth matches.

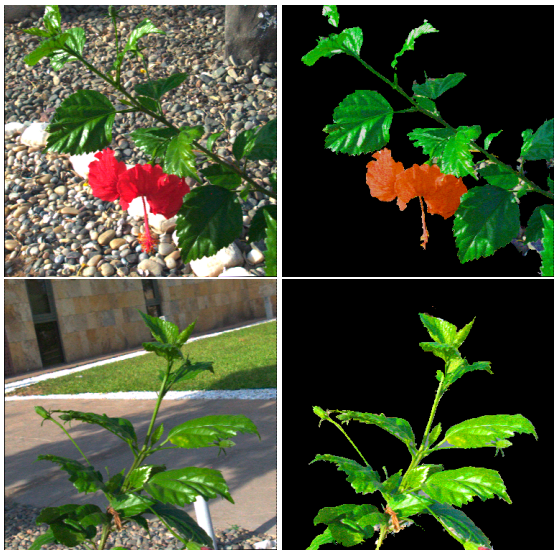


Fig. 19. User-guided segmentation. Regions comprising a single object are selected by a user. The regions themselves are not manually altered.

7.2 Light Field Flattening

Image flattening refers to the suppression of texture detail while preserving strong scene edges and overall image structure. Here, we extend an existing 2D method [Bi et al. 2015] to 4D. Specifically, we take into consideration the L_1 sparsity in spatial slices, angular patches as well as the 4D light field segmentation, and jointly minimize the pixel variation and approximation error as detailed in the following.

Spatial Term. f_i is the *Lab* feature vector of pixel p_i.

$$E_l = \sum_i \sum_{p_j \in N_h(p_i)} w_{ij} \|L(i) - L(j)\|_1, \tag{23}$$

where $N_h(p_i)$ is a spatial local $h \times h$ patch. w_{ij} is the affinity between pixel p_i and p_j. Here, we simply use Euclidean distance with a normalization function.

Angular Term. We prefer a uniform intensity values over simple angular patches of the light field, and smooth exposure variation in different spatial slices. Similar to Eqn. 23, we formulate our angular flattening term as

$$E_a = \sum_i \sum_{p_j \in N_a(p_i)} w_{ij} \|L(i) - L(j)\|_1, \tag{24}$$

where $N_a p_i$ is the angular patch that p_i lies in.

Segmentation Term. The segmentation provides extra cues to include more pixels for avoiding the influence of shading, reflectance or noise.

$$E_s = \sum_{p_i \in s_k} \sum_{p_j \in s_k} w_{ij} \|L(i) - L(j)\|_1, \tag{25}$$

Data Fidelity Term. To avoid trivial solution, smoothed light field should be similar to original light field,

$$E_d = \sum_i \|L(i) - L^{ini}(i)\|_2, \tag{26}$$

where L^{ini} is original light field data. The overall objective function is the sum of those terms,

$$E = E_d + \alpha E_l + \beta E_a + \gamma E_s, \tag{27}$$

where α, β, γ are weight parameters. Fig. 20 shows the results of the light field segmentation, where we then utilize the segmentation cue to remove fine details and preserve the main edges of the light field. In the example of Fig. 21, we visualize our light field segmentation, edge detection results and pencil sketching. We first utilize our light field segmentation for removing fine details of light field. Then, we apply conventional edge detection method [Dollár and Zitnick 2015] on smoothed light field to capture main edge. Light field segmentation provide a closure and compact region cues from light field flattening, yields a larger range smoothing. The removal of details texture shows light field abstraction with more clean edge, which forms a art-composition style of pencil sketching.

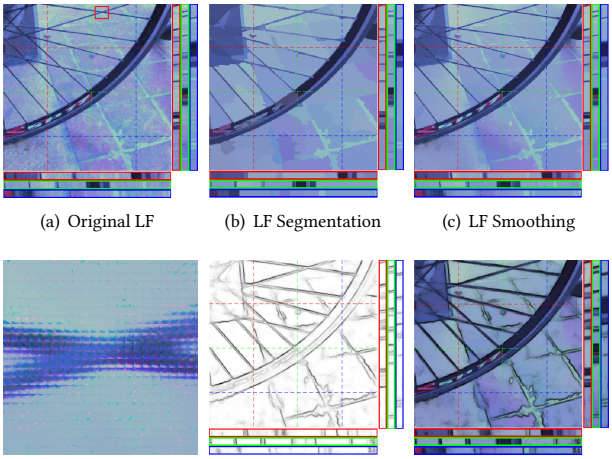
7.3 User-Guided Refocus Enhancement

A well-known and often used feature of light fields is the ability to refocus at different scene depths. The main target at a user-specified depth is sharp, while other depth ranges will be blurry. However, the adjustment of refocus depth is limited by the physical aperture of the light field camera, and therefore the level of blur cannot exceed a certain value. We combine light field segmentation with refocusing, and propose a way to enhance the blur, which is described as

$$L_\alpha(x', y', u, v) = L(u + \frac{x' - u}{\alpha(x', y')}, v + \frac{y' - v}{\alpha(x', y')}, u, v), \tag{28}$$

(a) Original (b) LF L_1 Flattening

Fig. 20. Light field flattening results.



(a) Original LF (b) LF Segmentation (c) LF Smoothing

(d) Angular Patch (e) Pencil Sketching (f) Abstraction

Fig. 21. The illustration of light field flattening and pencil sketching. (a) is original light field. (b) is our light field segmentation, we visualize segmentation as mean color of regions. (c) is angular patch of smoothed light field. (d) example light field angular patch of red rect in (a). (e) pencil sketching rendering on smoothed light field. (f) light field abstraction and edge enhancement.

$$\alpha(x', y') = \alpha \times m(x', y'), \tag{29}$$

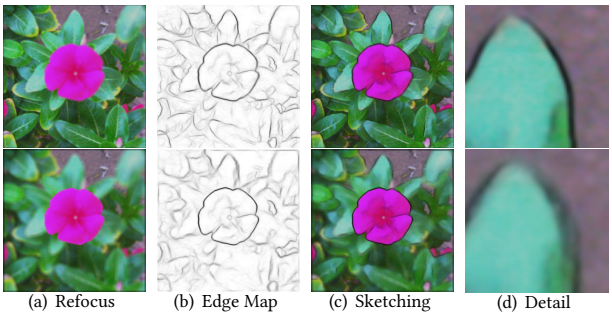
$$m(x', y') = \begin{cases} 1, \text{if } (x', y') \text{ is masked,} \\ k, \text{if } (x', y') \text{ is not masked.} \end{cases} \tag{30}$$

where $m(x, y)$ is a mask generated by user. For user-specified objects, we apply the original refocus function, for the background, we enhance the refocus ratio by multiplying extra parameters k. Fig. 22 shows our user-guided refocusing results. Our method provides more blur for background when comparing the naive refocusing method.

(a) Naive Refocus (b) User-guided Refocus

Fig. 22. The illustration of user-guided refocus. (a) Naive refocus, (b) user selection of light field segmentation.

User-Guided Edge Enhancement. In a similar fashion, we can highlight edges on user-defined objects. Fig. 23 shows the results of user-guided edge detection and pencil sketching. Our user-guided edge detection weaken the significant edge on unmasked area, yields a more significant edge for user-specified region.



(a) Refocus (b) Edge Map (c) Sketching (d) Detail

Fig. 23. User-guided edge enhancement and sketching.

8 CONCLUSION

In this paper, we solve the 4D light field segmentation problem using a new depth and occlusion consistent appearance term in combination with a novel view consistency term. Crucially, the resulting energy function is submodular, and can therefore be optimized efficiently using a greedy heuristic approach. We combine this new light field segmentation concept with several innovations to solve submodular optimization problems on very large graphs hierarchically, and very efficiently in both memory and time. In the future, we will explore other applications of light field segmentation, as well as new uses of the hierarchical submodular optimization.

REFERENCES

Radhakrishna Achanta, Pablo Marquez Neila, Pascal Fua, and Sabine Süsstrunk. 2018. Scale-Adaptive Superpixels. *Color and Imaging Conference Final Program and Proceedings* (2018).

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *PAMI* (2012).

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. 2011. Contour Detection and Hierarchical Image Segmentation. *PAMI* (2011).

Alper Ayvaci and Stefano Soatto. 2012. Detachable Object Detection: Segmentation and Depth Ordering from Short-Baseline Video. *PAMI* (2012).

Sai Bi, Xiaoguang Han, and Yizhou Yu. 2015. An L1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *TOG* (2015).

Jiansheng Chen, Zhengqin Li, and Bo Huang. 2017. Linear Spectral Clustering Superpixel. *TIP* (2017).

Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. 2017. SegFlow: Joint Learning for Video Object Segmentation and Optical Flow. In *ICCV*.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*.

Donald G. Dansereau, Bernd Girod, and Gordon Wetzstein. 2019. LiFF: Light Field Features in Scale and Depth. In *CVPR*.

Donald G. Dansereau, Ian Mahon, Oscar Pizarro, and Stefan B Williams. 2011. Plenoptic flow: Closed-form visual odometry for light field cameras. In *IROS*.

Piotr Dollár and C Lawrence Zitnick. 2015. Fast edge detection using structured forests. *PAMI* (2015).

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. 2004. Efficient Graph-Based Image Segmentation. *IJCV* (2004).

Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. 2009. Class segmentation and object localization with neighborhoods. In *ICCV*.

Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. 2010. Efficient Hierarchical Graph Based Video Segmentation. In *CVPR*.

Matthieu Hog, Neus Sabater, and Christine Guillemot. 2016. Light Field Segmentation Using a Ray-Based Graph Structure. In *ECCV*.

Matthieu Hog, Neus Sabater, and Christine Guillemot. 2017. Super-rays for Efficient Light Field Processing. *IEEE Journal of Selected Topics in Signal Processing* (2017).

Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldlücke. 2016. A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields. In *ACCV*.

Aaron Isaksen, Leonard McMillan, and Steven J Gortler. 2000. Dynamically reparameterized light fields. In *SIGGRAPH*.

Naeemullah Khan, Marei Algarni, Anthony Yezzi, and Ganesh Sundaramoorthi. 2015. Shape-tailored local descriptors and their application to segmentation and tracking. In *CVPR*.

Naeemullah Khan, Byung-Woo Hong, Anthony Yezzi, and Ganesh Sundaramoorthi. 2017. Coarse-to-Fine Segmentation with Shape-Tailored Continuum Scale Spaces. In *CVPR*.

Andreas Krause and Daniel Golovin. 2014. Submodular Function Maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *SIGKDD*.

Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *SIGGRAPH*.

Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. 2014. Saliency detection on light field. In *CVPR*.

Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. 2011. Entropy rate superpixel segmentation. In *CVPR*.

Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. 2014. Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint. *PAMI* (2014).

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Ehsan Miandji, Saghi Hajisharif, and Jonas Unger. 2019. A Unified Framework for Compression and Compressed Sensing of Light Fields and Light Field Videos. *TOG* (2019).

Hajime Mihara, Takuya Funatomi, Kenichiro Tanaka, Hiroyuki Kubo, Yasuhiro Mukaigawa, and Hajime Nagahara. 2016. 4D light field segmentation with spatial and angular consistencies. In *ICCP*.

Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. 2005. Light field photography with a hand-held plenoptic camera. (2005).

Sylvain Paris and Frédo Durand. 2007. A Topological Approach to Hierarchical Segmentation using Mean Shift. In *ICCV*.

Zhao Pei, Yanning Zhang, Tao Yang, Xiuwei Zhang, and Yee-Hong Yang. 2012. A novel multi-object detection method in complex scene using synthetic aperture imaging. *Pattern Recognition* (2012).

David Stutz, Alexander Hermans, and Bastian Leibe. 2018. Superpixels: An evaluation of the state-of-the-art. *CVIU* (2018).

Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. 2013. Depth from combining defocus and correspondence using light-field cameras. In *ICCV*.

Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. 2012. Seeds: Superpixels extracted via energy-driven sampling. In *ECCV*.

Kartik Venkataraman, Dan Lelescu, Jacques Duparré, Andrew McMahon, Gabriel Molina, Priyam Chatterjee, Robert Mullis, and Shree Nayar. 2013. Picam: An ultra-thin high performance monolithic camera array. *TOG* (2013).

Ting-Chun Wang, Alexei A Efros, and Ravi Ramamoorthi. 2015a. Occlusion-Aware Depth Estimation Using Light-Field Cameras. In *ICCV*.

Wenguan Wang, Jianbing Shen, and F. Porikli. 2015b. Saliency-aware geodesic video object segmentation. In *CVPR*.

Sven Wanner, Christoph Straehle, and Bastian Goldluecke. 2013. Globally Consistent Multi-label Assignment on the Ray Space of 4D Light Fields. In *CVPR*.

Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. 2005. High performance imaging using large camera arrays. *TOG* (2005).

Gaochang Wu, Belen Masia, Adrian Jarabo, Yuchen Zhang, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu. 2017. Light Field Image Processing: An Overview. *IEEE Journal of Selected Topics in Signal Processing* (2017).

Yichao Xu, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi. 2015. TransCut: Transparent Object Segmentation From a Light-Field Image. In *ICCV*.

Kaan Yücer, Alexander Sorkine-Hornung, Oliver Wang, and Olga Sorkine-Hornung. 2016. Efficient 3D Object Segmentation from Densely Sampled Light Fields with Applications to 3D Reconstruction. *TOG* (2016).

Guangming Zang, Mohamed Aly, Ramzi Idoughi, Peter Wonka, and Wolfgang Heidrich. 2018a. Super-Resolution and Sparse View CT Reconstruction. In *ECCV*.

Guangming Zang, Ramzi Idouchi, Ran Tao, Gilles Lubineau, Peter Wonka, and Wolfgang Heidrich. 2018b. Space-time tomography for continuously deforming objects. *TOG* (2018).

Guangming Zang, Ramzi Idoughi, Ran Tao, Gilles Lubineau, Peter Wonka, and Wolfgang Heidrich. 2019. Warp-and-project Tomography for Rapidly Deforming Objects. *TOG* (2019).

Hao Zhu, Qi Zhang, and Qing Wang. 2017. 4D Light Field Superpixel and Segmentation. In *CVPR*.