

## Repositório ISCTE-IUL

---

Deposited in *Repositório ISCTE-IUL*:

2021-12-07

Deposited version:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Hamad, M., Conti, C., Almeida, A. M. de., Nunes, P. & Soares, L. D. (2021). SLFS: Semi-supervised light-field foreground-background segmentation. In 2021 Telecoms Conference (ConfTELE). Leiria: IEEE.

Further information on publisher's website:

[10.1109/ConfTELE50222.2021.9435461](https://doi.org/10.1109/ConfTELE50222.2021.9435461)

Publisher's copyright statement:

This is the peer reviewed version of the following article: Hamad, M., Conti, C., Almeida, A. M. de., Nunes, P. & Soares, L. D. (2021). SLFS: Semi-supervised light-field foreground-background segmentation. In 2021 Telecoms Conference (ConfTELE). Leiria: IEEE., which has been published in final form at <https://dx.doi.org/10.1109/ConfTELE50222.2021.9435461>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

---

### Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

---

# SLFS: Semi-supervised Light-field Foreground-background Segmentation

Maryam Hamad  
Instituto Universitário de  
Lisboa (ISCTE-IUL),  
Instituto de  
Telecomunicações,  
Lisboa, Portugal  
maryam.hamad@lx.it.pt

Caroline Conti  
Instituto Universitário de  
Lisboa (ISCTE-IUL),  
Instituto de  
Telecomunicações,  
Lisboa, Portugal  
caroline.conti@lx.it.pt

Ana Maria de Almeida  
Instituto Universitário de  
Lisboa (ISCTE-IUL),  
ISTAR, Lisboa, Portugal,  
CISUC-Center for  
Informatics and Systems  
of the University of  
Coimbra  
ana.almeida@iscte-iul.pt

Paulo Nunes  
Instituto Universitário de  
Lisboa (ISCTE-IUL),  
Instituto de  
Telecomunicações,  
Lisboa, Portugal  
paulo.nunes@lx.it.pt

Luís Ducla Soares  
Instituto Universitário de  
Lisboa (ISCTE-IUL),  
Instituto de  
Telecomunicações,  
Lisboa, Portugal  
lds@lx.it.pt

**Abstract**—Efficient segmentation is a fundamental problem in computer vision and image processing. Achieving accurate segmentation for 4D light field images is a challenging task due to the huge amount of data involved and the intrinsic redundancy in this type of images. While automatic image segmentation is usually challenging, and because regions of interest are different for different users or tasks, this paper proposes an improved semi-supervised segmentation approach for 4D light field images based on an efficient graph structure and user’s scribbles. The recent view-consistent 4D light field superpixels algorithm proposed by Khan et al. is used as an automatic pre-processing step to ensure spatio-angular consistency and to represent the image graph efficiently. Then, segmentation is achieved via graph-cut optimization. Experimental results for synthetic and real light field images indicate that the proposed approach can extract objects consistently across views, and thus it can be used in applications such as augmented reality applications or object-based coding with few user interactions.

**Keywords**—light field segmentation, foreground-background segmentation, superpixels, graph-cut, semi-supervised segmentation

## I. INTRODUCTION

When humans look at images, their brains can easily classify the objects in the scene by distinguishing the object’s borders and understand the content. However, this task is much harder for computers which consider the scene as an array of pixels. To analyze the scene and understand its content by identifying meaningful objects, computers typically must start by applying image segmentation, which is the process of partitioning an image into smaller parts with homogenous properties. In computer vision, there are low-level, mid-level and high-level image segmentation techniques depending on the semantic meanings of the resulting segments. Basically, low-level image segmentation divides the image into smaller regions automatically with similar visual characteristics, such as color or depth, but not necessarily with a semantic meaning, and it can be used as a pre-processing step for object tracking or image editing [1], [2]. Mid-level image segmentation divides the image into a smaller number of larger regions (i.e., objects), it may be assisted with user interaction, however, it does not have semantic labels for the objects [3]. In addition to the mid-level segmentation output, the high-level image segmentation, can be assisted with high-level knowledge or learning process to



Fig. 1. Example of the proposed segmentation approach: a) a reference image with user’s foreground and background scribbles; b) the segmented object based on the scribbles.

obtain semantic meaning for the objects (e.g., a car, a flower, etc.) [4], which is out of this paper’s scope. In this paper, a combination of low-level image segmentation and user scribbles are considered to obtain mid-level (e.g., foreground-background segmentation) without having pre-defined semantic labels for the objects.

Although image segmentation is usually considered as a challenging problem, certain conditions can make it even harder, such as overlapping between objects with poor contrast or the huge amount of data, as in the 4D Light Field (LF) images, specifically when pixels are used as graph nodes. 4D LF images can be obtained by an array of cameras or by a single camera equipped with a special microlens array in front of the sensor or a moving camera gantry to capture different viewpoint images at different times. LF images record not only the intensity of light but also the angular direction of light rays [5]. The resulting 4D LF image, which can have a very large number of pixels, can be interpreted as a 2D array of 2D views and parametrized as  $L(x, y, u, v)$  where  $x, y$  are the spatial geometry of pixels in each view and  $u, v$  are the angular geometry of views. The 2D views are obtained from slightly different perspectives. While the 4D LF images contain a huge number of pixels, the similarity between pixels in different views can be used to reduce the computational complexity [1]. Furthermore, one of the most important advantages of 4D LF imaging is that it inherently includes depth information in its structure, which can be used in clustering and label propagation. In general, when traditional 2D segmentation is applied to 4D LF images, the information from adjacent views is not considered to resolve object occlusions, thus resulting in inconsistent segmentation across views. In order to cope with these challenges, the 4D LF image structure should be adequately considered. Various LF segmentation techniques have been proposed in the literature [3], [6]–[10]. However, most 4D LF segmentation techniques are either time-consuming, not interactive, not proposed for full consistent 4D LF segmentation or relying on accurate depth estimation.

To overcome the existing limitations and because the regions of interest are different for different users or tasks, an improved interactive Semi-supervised 4D LF Foreground-background Segmentation (SLFS) solution is proposed (see Fig. 1). This approach can be widely applied in object-based LF coding, augmented reality applications, or object extraction. Similar concepts to the segmentation algorithm proposed in [9], such as the graph-based image segmentation and the graph-cut optimization technique are used in this paper. However, different superpixel algorithm (i.e., the state-of-the-art View Consistent Light Field Superpixel (VCLFS) [10]) is exploited as graph nodes, enabling a dramatic reduction in the size of the graph and to effectively propagate the segmentation consistently across views, without the need for extra accurate depth estimation algorithm.

The remainder of the paper is organized as follows: Section II briefly reviews the related work on 4D LF image segmentation available in the literature; Section III describes the proposed approach in detail; Section IV evaluates the SLFS performance through a series of experimental results; Section V concludes the paper with some final remarks and proposes directions for future work.

## II. RELATED WORK

Image segmentation is a fundamental task in computer vision, and it has been attracting the attention of researchers for many years. Several image segmentation solutions for 2D images have already been proposed, however, only a few solutions have been proposed to tackle the 4D LF challenges, such as the huge amount of data and the need for ensuring the segmentation consistency across views. For low-level image segmentation, 4D LF superpixels/superrays have been proposed in [1], [8], [10] and can be used to enhance LF editing tasks (e.g., by propagating the edits into a 4D LF superpixel instead of a single pixel). For the case of mid-level image segmentation, Wanner et al. [3] proposed the first variational framework for multi-label segmentation, where the color and disparity cues of input seeds are used to train a machine learning classifier (i.e., random forest) that is used to predict the label of each pixel. However, the segmentation is not performed on the full 4D data (only the central view is segmented), the authors mentioned that the optimization step can take  $\sim 5$  minutes on a modern GPU if applied for all views. Mihara et al. [6] improved Wanner’s approach by building a graph in 4D space with spatial and angular neighbors and then using graph-cut for multi-label segmentation. Due to the huge number of graph nodes and the high computational time, only a fraction of the LF views (i.e.,  $5 \times 5$ ) were considered in the experiments. To reduce the graph size, Hog et al. [7] proposed a novel graph representation that utilizes the ray bundle (i.e., a set of all rays describing the same 3D scene point) as a graph node and exploited the redundancy in the LF data, decreasing the running time of the Markov Random Field (MRF) optimization and achieving entire 4D LF views segmentation. However, their approach depends on quite accurate depth estimation on all the views, thus, inaccurate individual depth maps greatly increase the running time and decrease the segmentation coherence. Additionally, the segmentation results can be very sensitive to the noise in real LF images.

It has been proven the efficiency of achieving mid-level and high-level segmentation based on low-level (e.g., superpixel) segmentation [2]. Lv et al. [9] recently proposed a novel hypergraph representation for 4D LF multi-label

segmentation by exploiting the superpixels proposed in [8] as hypernodes to reduce the graph size. However, Lv et al.’s approach relies on superpixel segmentation that requires depth estimation from extra algorithm, hence, it can be time-consuming. Additionally, it is not as accurate for real LF images as for the synthetic LF images due to the lack of accurate estimated depth map. Our approach is different from the recent work in [9], by replacing the used superpixels and simplifying the graph structure and size. Our approach is designed to interactively extract foreground from background similar to the recent work in 2D images [2], however, the segmentation is applied for all 4D LF data to achieve effective interactive segmentation of user’s region of interest.

## III. PROPOSED LIGHT FIELD SEGMENTATION APPROACH

In order to achieve foreground-background 4D LF image segmentation, the proposed approach consists of four major steps (see Fig. 2):

### A. LF superpixel extraction

In contrast to the widely used 2D superpixel algorithms, such as Simple Linear Iterative Clustering (SLIC) in [11], which divide an image into smaller clusters with similar visual appearance and spatial geometry, 4D LF image segmentation algorithms need to consider the depth information to extract consistent 4D LF superpixels. From the few proposed 4D LF superpixel algorithms, the state-of-the-art VCLFS algorithm is used in our proposed algorithm for the following reasons. Firstly, the VCLFS algorithm does not require an external depth estimation algorithm, since it implicitly estimates the disparity by computing the slopes of Epipolar Plane Image (EPI) lines for all LF views [10]. Secondly, the occluded objects where the foreground and background lines are intersected in the EPI are considered in the VCLFS algorithm and properly detected to prevent wrong segmentation. Finally, it outperforms other LF superpixel algorithms, notably [8], that is used in the recent 4D LF multi-label segmentation algorithm [9], in terms of boundary adherence, view consistency and running time [10], which is important for later foreground and background segmentation.

The VCLFS algorithm consists of three major steps: i) line extraction from the EPIs of central horizontal and vertical views of a 4D LF image; ii) occlusion-aware EPI segmentation; and iii) spatio-angular clustering by projecting the EPI segments of the central views into the central view and firstly clustering the central view using K-means algorithm, where the CIELAB color space, position and disparity are used. Afterward, the clustering labels are propagated across all views based on the EPI segments and disparity. After superpixels are extracted, the texture is characterized by using histograms of the superpixels’ intensities. To compute the histograms, the image is converted to the Hue, Saturation and Value (HSV) color space first. The HSV color space is designed to approximate the human vision perception and it is widely used for image analysis and segmentation [12]. To achieve luminance invariance, the value channel is not considered, and the histogram is computed using only the hue and saturation channels. For each superpixel, a 2D histogram of hue and saturation values is computed. Each superpixels’ histogram is normalized by dividing it by its sum. The obtained superpixels and the corresponding histograms will be used in the next step to create the graph representation.

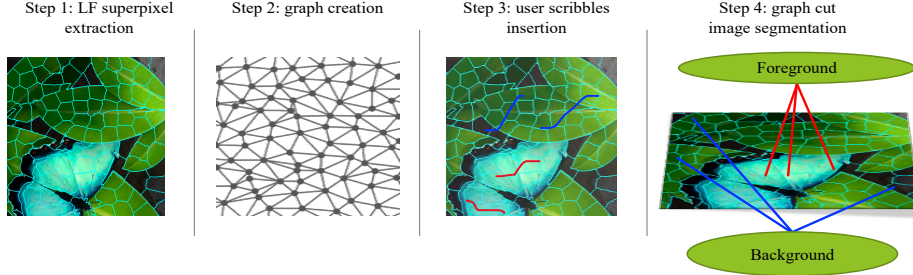


Fig. 2. Overview of the proposed SLFS algorithm: step 1) LF superpixels are extracted using the VCLFS algorithm; step 2) a graph is created using superpixels as graph nodes; step 3) scribbles are inserted by the user to initially label foreground and background superpixels; step 4) a graph-cut optimization is performed to the central view and propagated to the entire 4D LF views to iteratively achieve interactive foreground-background segmentation.

### B. Graph creation

Since our goal is to improve the 4D LF segmentation, the theory of graphs can be applied similarly to what has been done for 2D image segmentation. However, in the context of the 4D LF segmentation, several algorithms used a graph representation of the 4D LF image by representing each pixel as a graph node [6]. Due to the huge size of a LF image, the number of resulting graph nodes is also massive, leading to a high computational complexity not suitable for 4D LF interactive applications. In contrast, the hypergraph concept which is conceptually defined and used in [9] is similarly used in our approach and significantly reduces the graph size by defining the extracted 4D LF superpixels as graph nodes, however, we did not consider the angular neighbors or the multiple-target nodes as in [9]. Generally, a hypergraph is one type of graph representation that uses a set of nodes as one hypernode as well as the connected edges between two hypernodes as one hyperedge (see Fig. 3). Additionally, the hypergraph is coarsened into a planar graph by considering all corresponding superpixels across views as one hypernode.

In our graph representation, a planar graph is created on the central view superpixels and conceptually represented a hypergraph, where each hypernode in the central view includes all corresponding superpixels across views. The central view is chosen for two reasons: i) in dense 4D LF images, there is only a slight shifting across views and according to the Lambertian assumption, the 3D point of the scene is corresponding to a straight line in the EPI [10]. Thus, most superpixels in the central view having corresponding superpixels in all LF views with small disparities; and ii) the user is usually interested in segmenting frontal objects instead of small occluded objects. The corresponding superpixels across views are computed in the VCLFS by changing the spatial position of the central view superpixels based on the angular location of the view and the superpixels' disparities, and it assigns a same numeric label to the corresponding superpixels. The final segmentation will be propagated by assigning the corresponding superpixels across views, the same foreground or background labels as central view superpixels. In Fig. 3, a simplified hypergraph illustration is shown. In the red rectangle, there is an edge between two superpixels, similarly, the red edge exists in all 4D LF views in Fig. 3. The hypernodes  $S_i, S_j$  can be shown in the two circles below and connected with a hyperedge. In order to represent a graph, we need to define the edges between the graph nodes

and compute their weights. Since superpixels' shapes are irregular in most situations, the Delaunay Triangles algorithm<sup>1</sup> [13] is used to find the graph edges between neighboring superpixels' centroids. The Delaunay algorithm provided in the open-source Python library Sci-Py [14], [15] is used here.

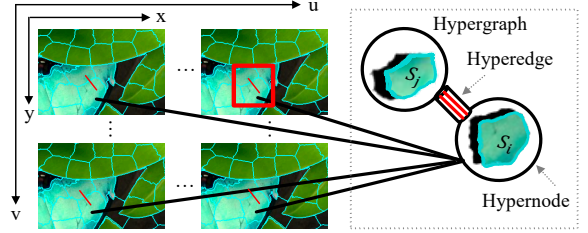


Fig. 3. The hypergraph representation where all corresponding superpixels across views are represented as one hypernode as in  $S_i$  and  $S_j$ . The red lines represent edges between two neighboring superpixels and, similarly, all corresponding edges between two hypernodes are represented as one hyperedge.

To create the graph  $G$  and perform graph-cut optimization to achieve foreground and background segmentation, the LF superpixels are used as nodes of the graph. Furthermore, two target nodes are added to the graph, for the foreground  $T_f$  (source node) and the background  $T_b$  (sink node), respectively (see Fig. 2). The maximum flow from the source to the sink is determined by the bottleneck (i.e., the edges minimum cut). Additionally, two different edge types are defined: i) target edges (i.e., the edges between the superpixel and the target nodes); and ii) neighboring edges (i.e., edges between spatially neighboring superpixels). After defining the types of the nodes and edge, we build a graph  $G = (v, \epsilon)$  of the central view, where  $v$  represents both superpixels and target nodes, and  $\epsilon$  represents edges between nodes. Each edge between superpixels is weighted by comparing the adjacent histograms using average Kullback-Leibler Divergence (KLD) [16] to compute the relative difference between histograms as in (1):

$$\mathcal{W}(S_i, S_j) = \mathcal{W}(S_j, S_i) = \lambda - \frac{1}{2} \left( \sum_x H_i(x) \log \left( \frac{H_i(x)}{H_j(x)} \right) + \sum_x H_j(x) \log \left( \frac{H_j(x)}{H_i(x)} \right) \right), \quad (1)$$

where  $H_i(x)$  and  $H_j(x)$  are, respectively, the hue and saturation 2D histograms of spatially adjacent superpixels  $S_i$  and  $S_j$  in the central view (as a complexity tradeoff in this

<sup>1</sup> The Delaunay algorithm finds a subdivision of a set of points into a non-overlapping set of triangles, such that no point is inside the circumcircle of any triangle.



paper, summations are over 20 histogram bins), and  $\lambda$  is a control parameter that helps in the graph-cut optimization process (after extensive testing, in our experiments a default value of  $\lambda = 25$  was used since it led to the best subjective results); this parameter is especially useful in case of very small or null difference between the superpixel histograms.

### C. User scribbles insertion

For semi-supervised interactive segmentation, a user can insert different scribbles to indicate the region of interest on the reference view. In this paper, the central view is selected as a reference view, since almost all views contain central view content with slight shifting. All superpixels under the scribbles are labeled either foreground or background, according to the scribble's label, and utilized as initial seeds to label unlabeled superpixels in the graph-cut step, where the cumulative foreground and cumulative background histograms are used. When user scribbles are inserted, graph target edges between labeled superpixels and target nodes are generated. Considering a superpixel under foreground scribbles, the edge weights between superpixel node  $S_f$  and the target nodes  $T_f$  and  $T_b$  represent the self-penalty  $D_S$  (i.e., the cost of labelling each superpixel as either foreground or background) as in (2) and (3):

$$D_{S_f}(T_f) = D_{S_i}(0) = 0, \quad (2)$$

$$D_{S_f}(T_b) = D_{S_i}(1) = W_{max}, \quad (3)$$

where  $D_{S_f}(T_f)$  is the edge weights between the foreground labeled superpixel and  $T_f$  (labeled as zero), and  $D_{S_f}(T_b)$  is the edge weights between the foreground labeled superpixel and  $T_b$  (labeled as one). A small value is assigned for foreground target edge if the superpixel is under foreground scribbles, while a high value is assigned for the background target edge to increase the self-penalty. In our experiments we fixed  $W_{max}$  to 100 as a high value. The same approach is used for superpixels under background scribbles.

### D. Graph-cut image segmentation

Generally, image segmentation can be formulated as the minimization of an energy cost function with two additive terms: i) the self-penalty (a.k.a data cost); and ii) the neighboring penalty (a.k.a smooth cost). Self-penalty represents the cost of labelling each superpixel as either foreground or background. Furthermore, the neighboring penalty ensures that neighboring superpixels are smooth and penalizes neighbors that have different labels.

To achieve the segmentation, graph-cut optimization is used, which is effective and handles image segmentation in terms of energy minimization [9]. The cumulative foreground and background histograms ( $H_{CF}, H_{CB}$ ) of the superpixels under the user scribbles are computed separately after the user's insertion. In order to assign a label for each unlabeled superpixel, the KLD is used to compute the relative difference between cumulative target histograms and the superpixel histogram as in (4):

$$D_{S_i}(T_{f \text{ or } b}) = \sum_x H_{CF \text{ or } CB}(x) \log \left( \frac{H_{CF \text{ or } CB}(x)}{H_i(x)} \right), \quad (4)$$

where  $D_{S_i}(T_{f \text{ or } b})$  is the self-penalty, and  $H_{CF \text{ or } CB}$  is foreground or background cumulative histogram. Suppose  $L$  is a label vector, which includes foreground (0) and background (1) labels for all the  $N$  superpixels  $L \in \{0,1\}^N$ . The energy function is computed by summing the data cost and smooth costs for assigning label  $l_i$  to superpixel  $S_i$  considering the labels of the neighbors  $\mathcal{N}_i$  as in (5) [17]:



$$E(L) = \sum_{S_i \in I} D_{S_i}(l_i) + \sum_{(i,j) \in \mathcal{N}_i} \mathcal{W}(S_i, S_j) |l_i - l_j|. \quad (5)$$

Finally, the graph-cut algorithm is applied to minimize the energy function to obtain the segmented result  $\mathcal{S}$  as in (6):

$$\mathcal{S} = \arg \min_L E(L), \quad (6)$$

where the energy function  $E(L)$  is the cost of assigning label  $l_i$  to each superpixel  $S_i$  in the image  $I$  by summing the data cost and the smooth cost for each superpixel  $S_i$  and its spatially neighboring superpixels  $S_j$ , where  $\mathcal{N}_i$  is the set of neighboring superpixels of  $S_i$ . In our algorithm, we take advantage of the optimized PyMaxFlow library to apply the graph-cut that implements the algorithm in [17] for central view. Since each superpixel in the central view conceptually represents a hypernode of all self-similar superpixels across views, the superpixels' labels from the central view are propagated to the entire 4D LF views by assigning each superpixel related to the hypernode to the label of the superpixel in the central view. The graph-cut optimization is interactively continued after each user's scribble insertion of both foreground and background scribbles, and the calculation of the cumulative target histograms are updated until the object segmentation is achieved according to the user's decision. Finally, the border's noise is removed from the final mask using median filtering with kernel size of  $(7 \times 7)$  and simple morphological operation (i.e., opening), with kernel size of  $(3 \times 3)$ . The used filters may slightly affect the spatial accuracy, but visually obtain smoother boundaries and reduce the noise.

TABLE I. IMAGE DATASETS USED IN THE EXPERIMENTAL RESULTS

4D LF image dataset	View resolution ( $x \times y$ ) pixels	Number of views	Thumbnail
HCI benchmark dataset [18]: <i>Papillon, Monasroom, Still life, Horses and Buddha</i>	768×768 pixels, except for horses: 1024×576 pixels	9×9	
EPFL MMSPG LF images dataset [19]: <i>Friends 4, Sphynx, and Sophie and Vincent 3</i>	625×434 pixels	15×15	

## IV. EXPERIMENTAL RESULTS

To evaluate the proposed approach, we implemented the proposed SLFS algorithm on a macOS computer with Intel i5 2.3 GHz processor and 8GB LPDDR3 memory. We used both synthetic 4D LF images [18] and 4D LF data captured with a Lytro Illum camera [19] as shown in Table I. The algorithm is implemented using Python programming language and the

open-source code for the VCLFS algorithm [20] was used to extract the 4D LF superpixels. The segmentation results are presented in Fig. 4 and Fig. 5, for synthetic and real LF images, respectively. Several parameters can affect the segmentation result, such as the superpixel size and image texture complexity. In the VCLFS algorithm, the segmentation size of  $x$  will generate average superpixel size of  $x^2$  pixels per superpixel (assuming a square shape) [20]. In our experiments (see Fig. 4 and Fig. 5), we set the segmentation size of the VCLFS to 30, to generate an average superpixel size of 900 pixels. This size of superpixel generates consistent and accurate segmentations with a reasonable computational complexity. In Fig. 6, we changed the size of superpixels to study its effect on segmentation. Larger sizes make the segmentation faster in terms of graph-cut optimization. However, it results in inaccurate segmentation due to the larger clusters that cannot be divided. On the other hand, smaller sizes result in a more accurate segmentation, but will increase the graph size and complexity. In Fig. 6, the segmentation graph-cut takes around 8 ms, when using VCLFS with a segmentation size of 100, but it takes around 35 ms and 82 ms for a segmentation size of 30 and 15, respectively. According to the image texture, images with complex texture require more scribbles than those with non-complex texture. In Fig. 4, the *Monasroom* image presents a complex texture requiring more user scribbles and interaction than in the *Papillon* image.

To compare our results with the other 4D LF segmentation algorithms that target multi-label segmentation, we used all the published segmentation masks in [7]. We were not able to compare with the recent work in [9] since there is no published masks or open-source code for the algorithm, additionally, there is no enough implementation details to reproduce it. Furthermore, similar work targeting foreground-background segmentation has been proposed for 2D images [2], and its comparison here would be unfair due to the 4D LF structure and propagation consistency. To enable the comparison with multi-label segmentation, we considered the targeted object (e.g., the yellow horse in Fig. 7) as a foreground and other labeled objects as background, hence, binary masks from the segmentation masks in [7] and the HCI segmentation ground truth in [18] are generated instead of multi-label masks. The comparison results are displayed in Table II, we used test images and their ground truth from the HCI dataset [18].

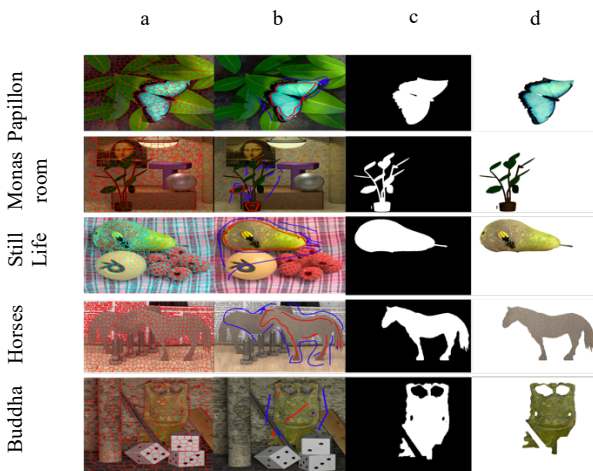


Fig. 4. SLFS results on the HCI dataset: a) central view with superpixels; b) user's foreground and background scribbles (blue for background and red for foreground); c) segmentation mask after graph-cut optimization; d) the segmented object.

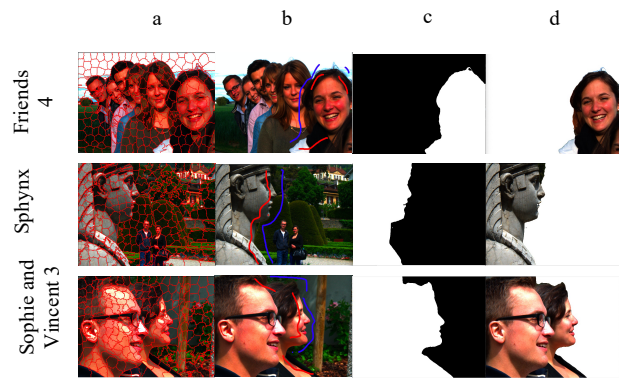


Fig. 5. SLFS results on the EPFL MMSPG dataset: a) central view with superpixels; b) user's foreground and background scribbles; c) segmentation mask after graph-cut optimization; d) the segmented object.



Fig. 6. Segmentation results for different superpixel's sizes: a) size = 100; b) size = 30; c) size = 15; (larger superpixels may create inaccurate segmentation results due to the larger cluster that cannot be divided while very small superpixels improves the accuracy and increases the graph complexity).

By using the hypergraph concept with the VCLF superpixels to represent the 4D LF image, a significant reduction in graph size is achieved. For example, the *Buddha* image has  $4.77 \times 10^7$  pixels, the algorithm in [7] reduced the graph size to  $8.19 \times 10^5$  nodes. Additionally, the algorithm in [9] reduced the graph size to  $1.46 \times 10^3$  nodes, while our algorithm reduced the graph size to only 625 nodes with similar accuracy as in Table II. Additionally, the segmentation result is consistent across views and adhere to the object's boundaries. Fig. 8, shows the consistent segmentation results, where our results show better visual consistency in some parts (e.g., the horse's hoof) compared to [7]. The VCLFS algorithm takes  $\sim 250s$  and  $\sim 273s$  for HCI and EPFL datasets respectively for superpixels extraction with superpixel size of 30, the graph-cut for the central view takes  $\sim 35ms$  and the propagation to all LF views takes  $\sim 3s$ .

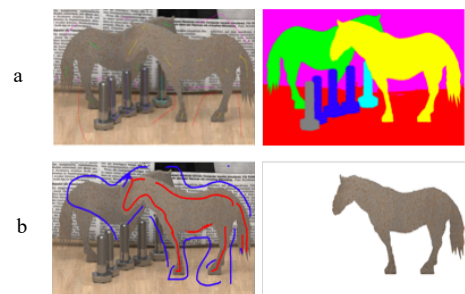


Fig. 7. Results from different interactive segmentation algorithms for *Horses* LF image: a) the state-of-the-art multi-label 4D LF segmentation result [9]; b) SLFS foreground-background segmentation result.

TABLE II. ACCURACY RESULTS FOR THE DIFFERENT ALGORITHMS UNDER ANALYSIS FOR VARIOUS 4D LF TEST IMAGES

	Results of [7]	Results of SLFS
Papillon	99.86%	99.66%
Still life	99.89%	99.87%
Horses	99.95%	99.59%
Buddha	99.57%	99.34%
Average	99.82%	99.62%

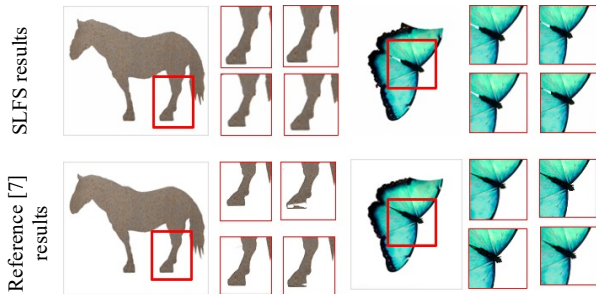


Fig. 8. Results from different 4D LF segmentation algorithms for *Papillon* and *Horses* 4D LF images. These images are selected to show the consistency across views where *Papillon* has uniform colors and *Horses* has complex texture. For each image: a red rectangle on the central image and zoomed patches from the top-left view, top-right view, bottom-left view and bottom-right view are shown, however, all the 4D LF views are segmented.

## V. FINAL REMARKS

In this paper, an improved interactive 4D LF foreground-background segmentation solution – *SLFS* – is proposed and evaluated. Firstly, the 4D LF superpixels are extracted efficiently using the VCLFS algorithm. Afterward, a hypergraph based on superpixels is created. Then, the segmentation problem is treated as an energy function optimization where a graph-cut technique is applied to optimize the segmentation result. Finally, the segmentation result is propagated to all 4D LF views consistently.

Experimental results were conducted on both real and synthetic 4D LF images and show the effectiveness of the proposed approach with comparable results even after the dramatic reduction in the graph complexity. Additionally, the experimental results show that the segmentation can be affected by the superpixel size, the image complexity and the graph-cut parameters.

The proposed approach can be used in several interesting applications where object extraction is needed, such as augmented and mixed reality, and object-based coding. For future work, the best compromise superpixel size to be used for this algorithm and the optimal parameters for graph creation and segmentation could be further optimized and will be considered. Additionally, the graph structure can be used for other LF editing applications, such as inpainting where the space after object extraction can be filled consistently by novel pixels in one view and propagated to the 4D LF views. Furthermore, this algorithm can be further improved to include the segmentation of the sparse 4D LF images where the nodes of the large occluded objects are handled particularly in the graph creation step.

## REFERENCES

- [1] M. Hog, N. Sabater, and C. Guillemot, "Superrays for Efficient Light Field Processing," *IEEE J. Sel. Topics Signal Processing*, vol. 11, no. 7, pp. 1187–1199, Oct. 2017.
- [2] W. Yu, Z. Hou, P. Wang, X. Qin, L. Wang, and H. Li, "Weakly supervised foreground segmentation based on superpixel grouping," *IEEE Access*, vol. 6, pp. 12269–12279, Feb. 2018.
- [3] S. Wanner, C. Strachle, and B. Goldluecke, "Globally Consistent Multi-label Assignment on the Ray Space of 4D Light Fields," in *2013 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Portland, OR, USA, June 23–28, 2013, pp. 1011–1018.
- [4] S. Jegou, M. Drozdal, D. Vazquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," in *2017 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, USA, July 21–26, 2017, pp. 1175–1183.
- [5] M. Levoy and P. Hanrahan, "Light field rendering," in *23rd annual conf. on Computer graphics and interactive techniques*, NY, USA, Aug. 1, 1996, pp. 31–42.
- [6] H. Mihara, T. Funatomi, K. Tanaka, H. Kubo, Y. Mukaigawa, and H. Nagahara, "4D light field segmentation with spatial and angular consistencies," in *2016 IEEE International Conf. on Computational Photography (ICCP)*, Evanston, IL, USA, May 13–15, 2016, pp. 1–8.
- [7] M. Hog, N. Sabater, and C. Guillemot, "Light Field Segmentation Using a Ray-Based Graph Structure," in *European Conf. on Computer Vision (ECCV)*, Amsterdam, Netherlands, Oct. 8, 2016, pp. 35–50.
- [8] H. Zhu, Q. Zhang, and Q. Wang, "4D Light field superpixel and segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 21–26, 2017, pp. 6709–6717.
- [9] X. Lv, X. Wang, Q. Wang, and J. Yu, "4D Light Field Segmentation from Light Field Super-pixel Hypergraph Representation," *IEEE Trans. Vis. Comput. Graph.*, early access, Mar. 2020, doi: 10.1109/TVCG.2020.2982158.
- [10] N. Khan, Q. Zhang, L. Kasser, H. Stone, M. H. Kim, and J. Tompkin, "View-Consistent 4D Light Field Superpixel Segmentation," in *IEEE/CVF International Conf. on Computer Vision (ICCV)*, Seoul, Korea, Oct. 27–Nov. 2, 2019, pp. 7810–7818.
- [11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [12] N. A. Ibraheem, M. M. Hasan, R. Z. Khan, and P. K. Mishra, "Understanding Color Models: A Review," *ARPN J. Sci. Technol.*, vol. 2, no. 3, pp. 265–275, Apr. 2012.
- [13] M. de Berg, M. van Kreveld, M. Overmars, and O. C. Schwarzkopf, "Delaunay Triangulations," in *Computational Geometry*, 2<sup>nd</sup> ed., Berlin, Germany: Springer, 2000, pp. 183–210.
- [14] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, Dec. 1996.
- [15] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020.
- [16] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.
- [17] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.
- [18] S. Wanner, S. Meister, and B. Goldlücke, "Datasets and Benchmarks for Densely Sampled 4D Light Fields," *VMV*, vol. 13, pp. 225–226, Sep. 2013.
- [19] M. Rerabek and T. Ebrahimi, "New Light Field Image Dataset," in *8th International Conf. on Quality of Multimedia Experience (QoMEX)*, Lisbon, Portugal, June 6–8, 2016.
- [20] N. Khan, Q. Zhang, L. Kasser, H. Stone, M. H. Kim, and J. Tompkin, "Repository for the ICCV 2019 paper: View-consistent 4D Light Field Superpixel Segmentation, by Khan *et al.*" [Online]. Available: <https://github.com/brownvc/lightfieldsuperpixels>. [Accessed: 30-Mar-2020].