IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 16, 2018, accepted February 11, 2018, date of publication February 27, 2018, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2810210

Weakly Supervised Foreground Segmentation Based on Superpixel Grouping

WANGSHENG YU[1], ZHIQIANG HOU[2], PENG WANG[1], XIANXIANG QIN[1], LIGUANG WANG[1], AND HUANYU LI[3]

[1]Information and Navigation College, Air Force Engineering University, Xi'an 710077, China
[2]School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[3]Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710051, China

Corresponding author: Wangsheng Yu (xing_fu_yu@sina.com)

This research was supported in part by the National Natural Science Foundation of China under Grant 61703423, Grant 61473309, Grant 61403414, and Grant 41601436, in part by the Natural Science Basic Reach Plan in Shaanxi Province of China under Grant 2016JM6050, and in part by the Aeronautical Science Foundation of China under Grant 20160196005.

ABSTRACT Image segmentation is of great significance to a variety of tasks in image processing and computer vision. Since fully unsupervised image segmentation is usually very hard in most cases, a task-oriented interactive segmentation approach becomes a popular solution. This paper proposes a weakly supervised image segmentation algorithm to extract foreground from a complex background relying only on a roughly predefined bounding-box. The algorithm integrates the Watershed algorithm and Mean-shift clustering algorithm to obtain reliable initial foreground and background labels for simple linear iterative clustering (SLIC) superpixels. Then, a synthetic superpixel grouping mechanism is proposed to group the remainder SLIC superpixels into foreground or background until the whole superpixels are completely grouped. The proposed algorithm reliefs the interactive information from users while maintaining the segmentation precision. Extensive experiments are performed, and the results indicate that the proposed algorithm can reliably segment the image foreground from the complex background with only a weakly supervision.

INDEX TERMS Computer vision, image segmentation, foreground segmentation, superpixel.

I. INTRODUCTION

Image segmentation is a fundamental problem in image processing and computer vision, which plays a significant role in object detection [1], recognition [2], tracking [3], etc. It is essentially a technology to group the image pixels into a set of regions, where each contiguous region with a delimited boundary is a superpixel. During the past decades, a lot of segmentation algorithms are proposed and quite a few famous works (such as ACM [4], Watershed [5], Mean-shift [6], GraphCut [7], ActiveCut [8], GrabCut [9], MSRM [10], simple linear iterative clustering (SLIC) [11], Turbopixels [12], Selective Search [2], SEEDS [13], etc.) pushed forward the development of image segmentation theory. Among these, the algorithms can be roughly divided into unsupervised approaches (Watershed [5], Mean-shift [6], SLIC [11], etc.) and supervised approaches (ACM [4], GraphCut [7], MSRM [10], etc.). Since the natural image likely contains very complex color and texture features, a totally unsupervised algorithm usually fails to obtain a satisfactory segmentation result [14]. Therefore, the approaches integrating the unsupervised low-level segmentation method and limited interactive inputs become popular in recent years.

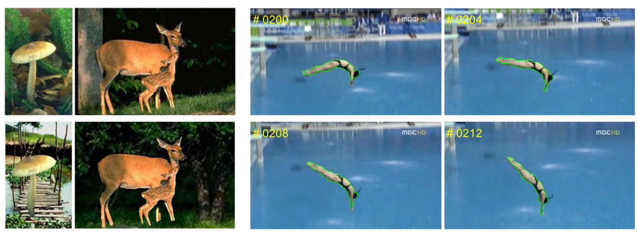


FIGURE 1. Some applications of the proposed algorithm: *left*, background change; *right*, video segmentation.

In this paper, we propose a weakly supervised image segmentation algorithm to extract foreground from a complex background, which can be widely applied in background change, video segmentation, etc. We refer to Fig.1 for more details. Our focus is to reduce the interactive inputs as much as possible while maintaining the segmentation performance. Generally, the prior knowledge provided by user may have great help to obtain an accurate result. However, it may also decrease the convenience and intelligence. So, we design our algorithm relying only on a roughly predefined bounding-box. Similar works can be found in GrabCut [9], Lazy snapping [15], MSRM [10], TouchCut [16], etc. With respect to the interactive information, the main difference is that GrabCut prefers to an input bounding-box that tightly encloses the foreground object, Lazy snapping and MSRM need both foreground and background labels, while TouchCut relies only on single stroke. Although TouchCut reliefs the dependence on user input, there exist deficiency when the foreground is complex. This mouse-clicks-based approach is also widely used in the semi-automatic segmentation of scattered and distributed objects [17] and image de-fencing algorithm [18]. Besides, some other meaningful works are proposed in recent years, such as Lazy Random Walks (LRW) [19], sub-Markov Random Walk (subRW) [20] and DBSCAN clustering algorithm [21] for superpixel segmentation, constrained Laplacian optimization [22] for interactive segmentation, and so on. The former three are different approaches to improve the performance of superpixel extraction and the fourth one is designed for interactive segmentation according to the user strokes. This interactive segmentation method is similar to MSRM [10] with respect to the interactive inputs. The main deficiency of these methods are the dependency on the user inputs and the sensitivity to those initial labels.

In order to reduce the interactive inputs while maintaining the segmentation performance, we select Watershed and SLIC to extract the preliminary superpixels and introduce Mean-shift clustering algorithm to learn reliable initial foreground and background labels according to the roughly input bounding-box. After that, a synthetic superpixel grouping (SSG) mechanism is proposed to partition the unlabeled superpixels into either foreground or background. We select Watershed and SLIC as preliminary superpixels for their effectiveness and efficiency. The Watershed superpixels are used to generate initial foreground and background labels and SLIC superpixels are used as inputs for grouping.

The remainder of the paper is organized as follows. Following a briefly literature review of related work, we describe the proposed algorithm in detail in Section 3. A comprehensive evaluation and the application discussion of the proposed method are provided in Section 4. A brief conclusion is given in the final section.

II. RELATED WORK

The past two decades have witnessed the emerging of image segmentation algorithms. A comprehensive literature survey of segmentation is beyond of the range of this paper. We only briefly review the tightly related works with respect to similar approaches, segmentation initialization and superpixel grouping.

A. SIMILAR APPROACHES

Foreground extraction is a high-level segmentation which can be implemented by grouping a set of over-segmented superpixels or contour evolution under a well-designed energy function. These interactive segmentation methods share a common operation that how to initialize the segmentation according to limited user inputs. Unlike digital matting that requires a cautiously labeled contour of the foreground, interactive segmentation seeks to alleviate the user's burden during the interaction while maintaining the segmentation accuracy. A natural solution is to give initial labels to both representative foreground and background via strokes of points or lines. Such a method appears in MSRM [10] and achieves a satisfactory result. This strategy can be further applied into GraphCut. One of the deficiencies is that the initial labels should cover almost the main features of the foreground and background. Another option is to substitute the strokes labels with a bounding-box tightly surrounding the foreground (such as GrabCut [9]). It has been pointed out that GrabCut algorithm fails when the bounding-box does not tightly cover the foreground object [19]. The preference of a tight bounding-box increases the burden of the human interaction, and moreover it prevents the algorithm from utilizing automatically generated bounding-boxes, such as boxes from object proposals. An improved version can be found in LooseCut [23]. A further alleviative work only requires a click of the foreground, which is known as TouchCut [16]. Within the TouchCut, it requires only a single finger touch to identify the object of interest in the image while the boundary of the whole image will be taken as background. However, this single touch approach has a limitation to make full use of the desired foreground, especially when the foreground is surrounded by cluttered background. In this paper, we compromise the single touch and tight bounding-box and propose an initialization scheme by combing loose bounding-box and clustering learning strategy. It owns the convenience like single touch while exploiting the prior knowledge as much as possible.

B. SEGMENTATION INITIALIZATION

Among the aforementioned approaches, segment the input image into over-segmented superpixels is a common preprocessing during the segmentation initialization. For example, the Mean-shift clustering algorithm is used in MSRM [10] to provide initialized segments. The authors also discussed another superpixels method as initial segments and compared the grouping results between these two. LooseCut [23] exploited a multiscale superpixels algorithm to cluster all the pixels into superpixels during the segmentation initialization step. Such superpixel algorithm considers both within-cluster feature consistency and the cluster-boundary smoothness for pixel clustering. Watershed and SLIC are very common in the unsupervised segmentation approaches, especially served as initial segmentations. Watershed is a very efficient segmentation approach based on mathematical morphology. There are two models (simulated immersion model and simulated rainfall model) to implement Watershed segmentation. It can obtain a connected, closed, stable and pixel-wise segmentation result. SLIC is another famous approach to obtain over-segmentation which adapts k-means clustering to generate superpixels while keeping its semantic meaning intact. It obtains stable and consistent segmentation in a very efficient manner. gSLIC [24] is an improvement of SLIC which runs at 250 *fps* for image with a resolution of 640 × 480. It makes SLIC further more efficient and suitable for real-time operation. Turbopixels is another fast superpixel extraction method based on geometric flows. It generates segments that on one hand respect local image boundaries, while on the other hand limit under-segmentation through a compactness constraint. The integration of Watershed and other superpixels theory is also an interesting research direction which can adopt the both advantages.

C. SUPERPIXEL GROUPING

Although some level-set based methods are directly applied on raw pixels under the driven of energy function, superpixel grouping is a very popular approach to obtain a high-level segmentation. Superpixel grouping, also known as region merge, has been researched for decades. A common scheme to treat with superpixel grouping is region adjacent graph (RAG) with each vertex stands for a region while each RAG edge for the similarity of two adjacent regions. The hard question is how to describe the similarity and design the merging strategy. Ning *et al.* [10] presented a simple but very effective grouping mechanism using the maximum similarity based on RGB color histogram. The grouping process is adaptive to the image content and it does not need to set the similarity threshold in advance. However, this method is sensitive to the initial markers in some cases. Within the selective search algorithm [2], the authors proposed a hierarchical grouping strategy to grouping the pre-segmented superpixels step by step. This hierarchical grouping strategy integrates four features (color, texture, size and fitness) and groups the superpixel with its most similar neighbor in each step until the whole image becomes a single region. In this paper, we seek to group the unlabeled superpixels into either foreground group or background group during the superpixel grouping stage, which is similar to MSRM. The kernel question is to measure the similarity of the unlabeled superpixels with labeled foreground and background. We propose a synthetic superpixel grouping (SSG) mechanism to consider both local and global similarities which achieves a pleasing result.

III. THE PROPOSED ALGORITHM

The proposed algorithm (demonstrated in Fig.2) consists of *superpixel extraction*, *label initialization*, *synthetic superpixel grouping* and *final segmentation*. We will introduce these contents one by one.

A. SUPERPIXEL EXTRACTION

There are several famous algorithms to extract the superpixels, such as Mean-shift, Watershed, SLIC, and so on. Among these algorithms, Watershed and SLIC are very common methods and widely applied as the input of some high-level segmentation algorithms. In the proposed algorithm,

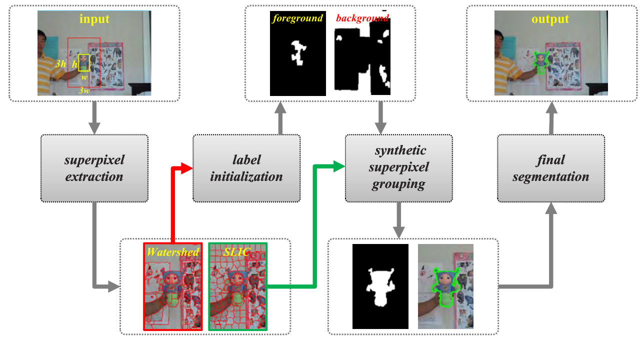


FIGURE 2. The flow chart of the proposed algorithm.

we select Watershed and SLIC as the low-level segmentation methods for the follow reasons. Firstly, as segmentation initialization, we want to cutdown the time-consuming as far as possible during the preprocessing. Both of these two methods are good choices because they are very efficient for superpixel extraction (millisecond-level for image patch with the size of 100 × 100). Secondly, these two algorithms have different emphasis. The Watershed algorithm rely more on globe information and can extract some weak edges. But it may produce over-segmentation in the texture areas. The SLIC algorithm is a local clustering algorithm to obtain superpixels. It can prevent over-segmentation according to the preset parameters. The integration of these two can make full use of both globe and local information. Thirdly, both of these two methods can be easily implemented for automatic process, which is of great significant for the proposed algorithm.

For Watershed segmentation, we introduce the marker-based Watershed algorithm which proposed by Yu *et al.* [25]. We slightly change the H-minima threshold initialization rule. In the original literature, this threshold is adaptively controlled by the mean intensity values of the corresponding water basin. In this paper, we sort the depth of the water basins descending and select the former N_1 water basins as the initial marker. The basic Watershed algorithm is then applied on the marker-based image to obtain the segmentation.

For SLIC segmentation, we directly introduce the original SLIC algorithm [11] to extract the superpixels. Although gSLIC is more efficient than the original SLIC, however, it has special requirements for hardware. The only parameter for SLIC superpixel algorithm need to be preset is the total number of superpixel regions. We preset this parameter as N_2. So, the Watershed algorithm will obtain N_1 regions and SLIC will also obtain approximately N_2 regions.

We exploit the Watershed algorithm to calculate the initial label of reliable foreground and background. The SLIC results are exploited for superpixel grouping. Note that we only do the low-level segmentation on the enlarged image patch (**three times** large as the input bounding-box in each direction) centered with the input bounding-box. We refer to Fig.2 for demonstration.

B. LABEL INITIALIZATION

In order to partition the superpixel regions into either foreground or background, we need an initial standard to distinguish these two types of regions. Here, we introduce the Mean-shift clustering algorithm to initialize the foreground and background labels.

Firstly, we give a label value for each pixel, of which the ones inside the input rough bounding-box are 1 and others are 0. Denote the Watershed regions as $\{r_k\}_{k=1,2,\cdots,K}$. We calculate label value for the k^{th} region as follows:

$$\hat{r}_k = \frac{\sum_{(x,y)\in r_k} \hat{p}(x,y)}{\sum_{(x,y)\in r_k} 1}, \tag{1}$$

where $\hat{p}(x, y)$ is the label value of pixel (x, y). If the pixel (x, y) is inside the input rough bounding-box, then $p(x, y) = 1$. Otherwise, $p(x, y) = 0$. Then, the Watershed regions are clustered via the Mean-shift clustering algorithm. We describe each Watershed region using a color histogram in HSI color space, which consists of hue, saturation and intensity channel. We further divide each channel into 8 bins so as to the image region can be described with a 512 (8^3) bins feature vector. Bhattacharyya coefficient [10] is employed to measure the similarity between two histogram vectors. The bandwidth of Mean-shift clustering is set as 0.25 in the proposed algorithm. We refer to the literature [6] for more details about this clustering course.

Denote the clustering result as $\{C_l\}_{l=1,2,\cdots,L}$. A similar processing, we can calculate the label value of the l^{th} cluster as follows:

$$\hat{C}_l = \frac{\sum_{(x,y)\in C_l} \hat{p}(x,y)}{\sum_{(x,y)\in C_l} 1}. \tag{2}$$

It is reasonable that each cluster and its inner superpixel regions should have the same label value. However, some superpixel region far from the foreground may also be clustered with the one falling into the foreground region when they have a near color distribution. To avoid this situation, we refine the label value of superpixel regions as follows:

$$\hat{r}'_i = \frac{\hat{r}_i + \hat{C}_j}{2}, \quad if: r_i \in C_j. \tag{3}$$

Based on the refined label value, we can select the reliable foreground label and background label. We sort the label values of superpixels according to the ascending order and select the largest m_1 as foreground label and the smallest m_2 as background label. Fig.3 demonstrates the initialized labels.

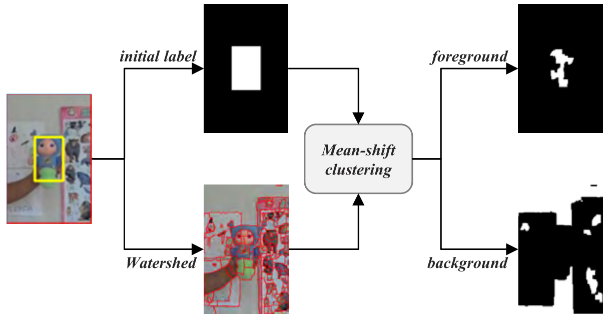


FIGURE 3. The demonstration of label initialization.

C. SYNTHETIC SUPERPIXEL GROUPING

We impose the labeled Watershed regions (foreground or background) onto the SLIC result. Based on this, some regions of the SLIC result are labeled as foreground and some regions are labeled as background. However, there are still some regions are unlabeled.

The main work of this step is to group the unlabeled regions into either foreground or background. To resolve this question, Ning *et al.* [10] proposed a maximal similarity based region merge algorithm (MSRM). They divided the whole course into two steps: background region merging with unlabeled region and merging between two unlabeled regions. These two steps repeat iteratively until no region can be merged any more. The remaining unlabeled regions are then classified into foreground. Based on a set of experiments, we find that this algorithm is sensitive to the input labels and may even produces a wrong segmentation result in some cases. Inspired by MSRM, we propose a synthetic superpixel grouping algorithm. The brief procedure can be described as follows:

Step 1: construct (update) the superpixels adjacent table (SAT), which describes the adjacent relationship and similarity between the superpixels. Here, the superpixels are described with histogram in HSI color space with 512 bins. The similarity metric is Bhattacharyya coefficient [10].

Step 2: for each unlabeled superpixel which is adjacent to the foreground or background, find out its most similar neighbor superpixel. If this superpixel is already labeled, give this label to the current superpixel. Otherwise, do nothing to the current superpixel. For each label change, the number of labeled superpixels increases by 1 while the number of unlabeled superpixels decreases by 1. This process will stop when there is no more label change.

Step 3: if there are still unlabeled superpixel regions, denote these regions as $\{r'_s\}_{s=1,2,\ldots,S}$. We define a synthetic superpixel distance (SSD) as follows to measure the distance between the unlabeled superpixel and the labeled foreground regions (RF) or background regions (RB):

$$SSD(r'_s, R) = 1 - \max\{sim(r'_s, R)\} \times mean\{sim(r'_s, R)\}, \tag{4}$$

where $sim(r'_s, R)$ is the function to compute the similarities between r'_s and the regions in $R = \{r_t\}_{t=1,2,\ldots,K}$ according to the following formula:

$$sim(r'_s, R) = \{\rho(r'_s, r_t)\}, \quad t = 1, 2, \ldots, T, \tag{5}$$

where $\rho(r'_s, r_t)$ is the Bhattacharyya coefficient, and T is the number of the regions in R.

If the current superpixel region satisfies the condition $SSD(r'_s, RF) < SSD(r'_s, RB)$, we group the current superpixel region into foreground. Otherwise, we group it into background.

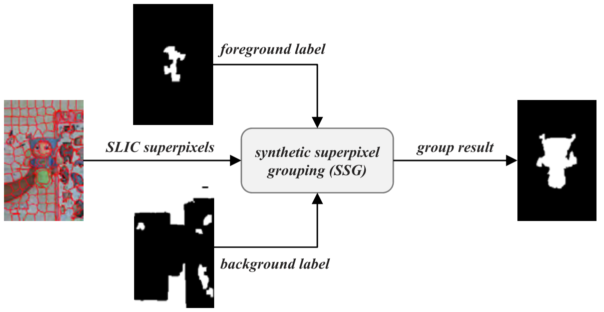


FIGURE 4. The demonstration of superpixel grouping.

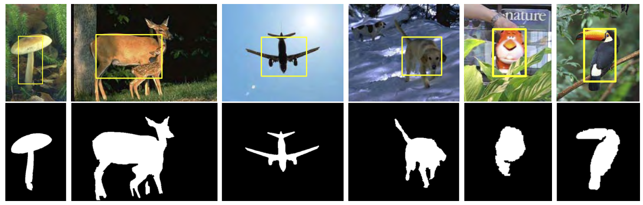


FIGURE 5. Some representative results of the proposed algorithm. The top row are the input images with loosely selected bounding-boxes, and the bottom row are the final segmentation results. The displayed images from left ro right are *fungus*, *deer*, *airplane*, *dog*, *tiger*, and *toucans* successively.

These three steps iterative until all the unlabeled superpixel regions are grouped into either foreground or background. After that, we obtain a binary marker image distinguishing the foreground and background. Fig. 4 demonstrates the course of superpixel grouping.

D. FINAL SEGMENTATION

Based on the initialized foreground and background label, the synthetic superpixel grouping can well group the unlabeled superpixel regions into either foreground or background. The rest image content outside the processed image patch is preset as background. So, the foreground is now well segmented out from the original image with only a weakly supervised input bounding-box. We refer to Fig.2 for a demonstration of the whole course.

IV. RESULTS AND ANALYSIS

In this section, we will comprehensively evaluate the proposed algorithm. After a brief introduction of parameters setting, some representative results of the proposed algorithm are provided. We then compare the segmentation performance of the proposed algorithm with some closely related state-of-the-art works. We further discuss the applications of the proposed algorithm in background change, video segmentation and visual tracking. Finally, the robustness analysis and faultiness analysis will also be considered in this section.

A. PARAMETER SETTINGS

The whole algorithm integrates the Watershed, SLIC, Mean-shift clustering and the proposed SSG. For Watershed and SLIC, the parameters are exploited to control the segmentation initialization. Denote the number of Watershed superpixels and SLIC superpixels as N_1 and N_2. Larger values will provide more precise initial segmentation, however, these will increase the time-consuming for subsequent processing. We preset $N_1 = 200$ and $N_2 = 200$ for the proposed algorithm after a number of experiments. We give the segmentation results under different parameters setting and analyze the influence to segmentation under different values of N_1 and N_2 in the subsequent content. When clustering the Watershed superpixels to obtain initial foreground and background labels, the feature dimension and bandwidth need to be preset for Mean-shift clustering algorithm. We use a histogram with 512 bins in HSI color space (8 bins for hue, saturation and intensity respectively) to describe the Watershed superpixels feature. This value is also used in SSG when describing the SLIC superpixels. The bandwidth of Mean-shift clustering is preset as 0.25. We found that a slightly change of bandwidth have very limited influence to the final result. After refining the label value, we need to select the largest m_1 as reliable foreground label and the smallest m_2 as reliable background label. Too small or too large values for these two parameters will decrease the segmentation precision. Based on a large number of tests, we find that the setting $m_1 = 10$ and $m_1 = 10$ produces relative better results. A slightly change of these two value may not introduce distinctive change for the final segmentation results. Unless mentioned otherwise, the aforementioned parameters are fixed for all the experiments.

B. REPRESENTATIVE RESULTS

Fig.5 displays some representative foreground segmentation results of the proposed algorithm. Considering the paper length restriction, we only select very limited results to demonstrate its effectiveness. Among the test images, some target foreground appear in relative clean background (such as the image *dear* and *airplane*), while others are integrated with cluttered background (such as the image *fungus* and *dog*). The difficulty is to segment out the foreground completely with restricted interactive information while suppress the background. We can see from Fig.5 that our algorithm obtains a pleasing result under a very loose bounding-box. Almost all the foreground pixels are grouped into the final segmentation while the background within the input bounding-box is well deleted. The segmentation is smooth and integrated.

C. COMPARISON TO SIMILAR WORKS

We also compared the proposed algorithm to some recent related works, such as MSRM [10] based on strokes input and LooseCut [23] based on loose rectangle input. MSRM is an efficient and effective algorithm to extract foreground which demonstrates a better performance than GraphCut [7]. LooseCut is a recent work which relies only a roughly input bounding-box. We select this algorithm as a comparison because it is also a rough input based method and owns a relative better performance than GrabCut [9]. For MSRM, we run the algorithm 5 times with cautiously initialized labels

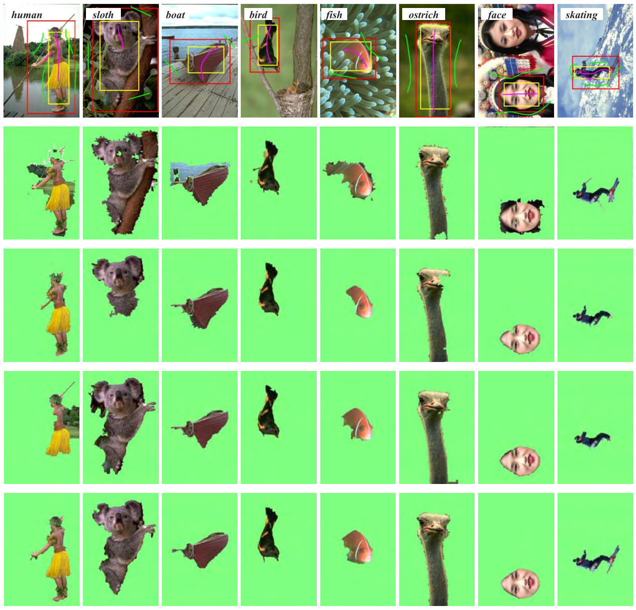


FIGURE 6. Segmentation results comparison with related works. From top to bottom, the original test images, the results of GrabCut [9], MSRM [10], LooseCut [23] and the proposed algorithm are successively displayed. In the top row, the red bounding-box is for GrubCut while yellow bounding-box is for LooseCut and the proposed algorithm. The initial labels for MSRM are purple (foreground) and green (background) strokes.

and selected a relative better result for each image. For LooseCut, we draw a large rectangle enough to enclose the whole foreground. As a baseline reference, the results of the original GrabCut [9] are also displayed.

The test images are selected from the Berkeley segmentation database (BSD) constructed by Martin *et al.* [26]. This database includes five hundred test images with human segmentation masks as the ground-truth. The subjective comparison of the segmentation algorithms is displayed in Fig.6. To be pointed out is that the GrabCut is heavily dependent on the input bounding-box, which must tightly enclose the foreground. Besides, MSRM needs some strokes to roughly label the foreground and background. In the top row of Fig.6, we displayed all the initialization information for the test algorithm. The red bounding-box is for GrubCut while yellow bounding-box is for LooseCut and the proposed algorithm. The initial labels for MSRM are purple (foreground) and green (background) strokes. It can be seen that GrabCut mistakes the background pixels for foreground pixels in some cases, such as the tree trunk when segmenting the *sloth*, and the lake surface when segmenting the *boat*. MSRM obtains a pleasing result when the background is simple (such as the image *boat* and *bird*) but the performance decreases when the background becomes cluttered (such as the image *sloth* and *skating*). LooseCut obtains relative better results than MSRM, however, the performance decreases when the discrimination between foreground and background decreases (such as the grass ring in the head and vegetation on the ground in test image *human*). Compared with the referenced approaches, the proposed algorithm behaves relative better in most of cases, especially in the test image *sloth* and *skating*.

In order to give a quantitative evaluation, we introduce two metrics [10] to compare the proposed algorithm with the referenced algorithms.

The first one is True Positive Rate (TPR) which is defined as the ratio of the number of correctly classified foreground pixels to the number of total foreground pixels in the ground-truth. Denote the binary images of algorithm result and ground-truth as A and B. The TPR value can be calculated as follows:

$$TPR = \frac{count(A \bigcap B)}{count(B)}, \tag{6}$$

where the $count(\cdot)$ is the function to count the total number of positive pixels in binary image, and $A \bigcap B$ is the intersection of A and B.

The second metric is the False Positive Rate (FPR). It is defined as the ratio of the number of background pixels but wrongly classified as foreground pixels to the number of background pixels in the ground-truth, which can be calculated as follows:

$$FPR = \frac{count(A \bigcap \overline{B})}{count(\overline{B})}, \tag{7}$$

where the $\overline{B}$ is the Non-logic operation of B, that is, the background of ground-truth.

We manually labeled the ground-truth pixels according to the database [26] for TPR and FPR values calculation. The comparison results are displayed in Tab.1. GrabCut obtains a higher TPR at the cost of increasing of FPR. Compared with MSRM and LooseCut, our method obtains a relative better performance in both TPR and FPR.

D. APPLICATION DISCUSSION

One of the main advantages of the proposed algorithm is that it can relief the user's burden while maintaining the segmentation performance. It is natural to apply it to some weakly supervised image segmentation areas. In this subsection, we mainly discuss the applications in *background change*, *video segmentation* and *visual tracking*.

1) BACKGROUND CHANGE

In many practical cases, we want to change the background of something what we are interested in. This technology is widely used in photo processing, post-production of films, and so on. In these situations, what we can be sure of is the foreground areas. Compared with the digital matting technology which relies much on the user's interactive information, the proposed algorithm can segment out the foreground in a very efficient way with only a loosely bounding-box input. In Fig.7, we display four examples to demonstrate the application of the proposed algorithm in background change. All the test images are selected from BSD database [26]. In the first image, we helped the News reporter change the background to an image with beautiful buildings. The foreground

TABLE 1. The TPR(%) and FPR(%) values of different algorithms on the test images.

test image	GrabCut [9]		MSRM [10]		LooseCut [23]		Ours	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
human	88.37	8.87	88.95	0.63	65.02	6.38	91.62	0.33
sloth	87.49	26.33	65.02	0.34	98.58	7.94	93.44	0.36
boat	96.92	8.44	98.72	2.31	96.30	0.60	96.91	0.27
bird	96.62	1.11	94.65	0.16	96.74	0.33	99.15	0.13
fish	94.55	4.86	71.88	0.04	89.98	0.12	92.56	0.11
ostrich	86.82	0.68	87.44	0.07	95.63	0.80	94.49	0.18
face	97.07	6.00	97.73	0.33	99.18	0.43	99.32	0.08
skating	92.64	1.84	75.60	0.01	71.35	0.13	92.07	0.24
Average	92.56	7.27	85.00	0.48	89.10	2.09	**94.94**	**0.21**

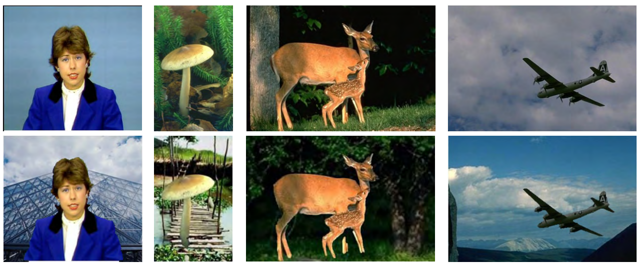


FIGURE 7. Background change using the proposed algorithm. The top row are the original images and the bottom row are the background change results. All the test images are selected from BSD database [26].

is well segmented out and integrated with the new background harmoniously. For the second image, we removed the fungus from jungle to a wooden bridge, which makes the image more interesting. In the rest two images, we also obtained two new images with no sense of violation. We can see from these examples that the proposed algorithm can be well applied into background change area and obtain a satisfactory result.

2) VIDEO SEGMENTATION

Video segmentation is a very interesting technology which is widely applied in video edit, video coding, video compression, and so on. Under normal conditions, video segmentation can be obtained by digital matting technology, however, it is too troublesome to label the initial foreground and background. With the help of the proposed algorithm, we can segment the foreground out from the video frames in a very efficient way. We can draw a loose bounding-box for each frame to finish the video segmentation, however, the most convenient and efficient way is to draw a initial bounding-box in the first frame and obtain the subsequent bounding-boxes via a tracker. Fig.8 displays the segmentation results of four video clips. All these video clips are collected from public available dataset. The first three are from VOT-2014[1] and the last one is provided by OTB-2015 [27]. During these experiments, we only draw a initial bounding-box for each video, the bounding-boxes of the rest frames are obtained

[1] Available at http://votchallenge.net

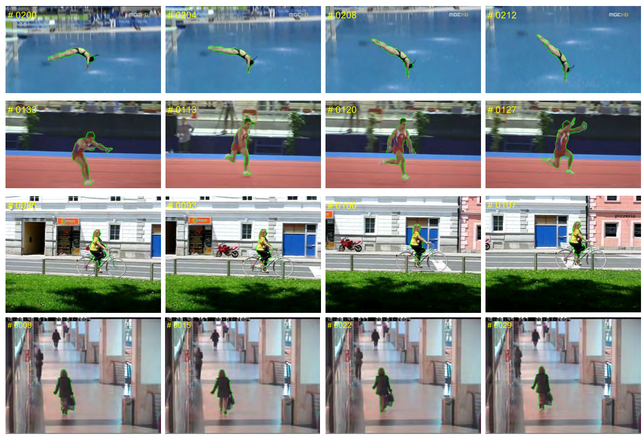


FIGURE 8. Video segmentation using the proposed algorithm. From top to bottom are the partial segmentation of the video clips ***diving***, ***gymnastics***, ***bicycle***, and ***walking***. Among these clips, the first three are from VOT-2014 and the last one is provided by OTB-2015.

by the Mean-shift tracker provided by OTB-2015 [27]. Some typical segmentation results are displayed in Fig.8. We can see that the proposed algorithm can well finish the video segmentation tasks when the background is not so much complicated.

3) VISUAL TRACKING

This proposed algorithm can be further applied to visual tracking task. Firstly, in some realistic cases, we can only obtain a rough location of the target rather than a cautiously selected bounding-box. In this case, we can refine the initial bounding-box using the proposed algorithm. Secondly, along with the tracking process, the tracking result may drift little by little. In this situation, we can correct the result using the proposed algorithm in every fixed number of frames. We select four representative video clips from OTB-2015 [27] to validate the effectiveness. We test the tracking performances using the Mean-shift tracking and the Mean-shift tracking with correction. It is known that Mean-shift is a very familiar tracking algorithm. It works well when the background is simple. However, the performance becomes worse when the contrast between foreground and background decreases. In this experiment, we first run the original Mean-shift tracking algorithm through the test

TABLE 2. The TPR(%) and FPR(%) values under different parameter settings of N_1 with $N_2 = 200$.

test image	$N_1 = 160$		$N_1 = 180$		$N_1 = 200$		$N_1 = 220$		$N_1 = 240$	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
human	89.61	1.34	91.04	0.38	91.62	0.33	91.62	0.33	90.12	0.47
sloth	91.29	0.84	93.44	0.36	93.44	0.36	92.71	0.48	87.71	1.41
boat	92.93	0.58	95.83	0.29	96.91	0.27	93.88	0.41	89.42	1.17
bird	98.11	0.31	98.82	0.21	99.15	0.13	99.15	0.13	99.03	0.17
fish	90.67	0.38	93.01	0.10	92.56	0.11	89.49	0.32	86.99	0.92
ostrich	88.74	0.93	94.38	0.28	94.49	0.18	92.43	0.36	89.54	1.03
face	96.14	0.25	99.32	0.08	99.32	0.08	99.32	0.08	97.28	0.22
skating	88.63	0.92	91.69	0.21	92.07	0.24	91.54	0.40	86.66	1.41
Average	92.02	0.69	94.69	0.24	**94.95**	**0.21**	93.77	0.31	90.84	0.85

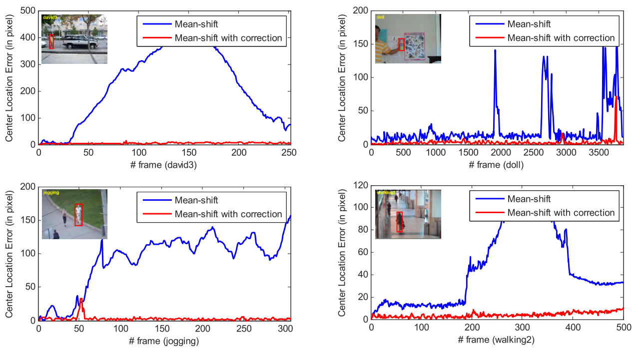


FIGURE 9. Tracking results comparison between the original Mean-shift tracking and the Mean-shift tracking with correction. In the top-left corner of each subplot, a red bounding-box shows the initial target to be track.

videos, and then run the Mean-shift algorithm integrated with a correction processing based on the proposed algorithm. The current tracking result will be used as a loose bounding-box for the following frame and the segmentation result will be used to correct the tracking drift. This correction process is added every three frames. We record the tracking results of both Mean-shift and Mean-shift with correction and compared the Center Location Error (CLE) curves in Fig. 9. CLE is a metric to evaluate the tracking performance, which measures the error between the centre locations of tracking window and the ground-truth. For the sequence *david3*, the average CLE value of Mean-shift tracking is 222.4, while this value decreases to 5.6 when adding a correction processing. For the sequences *doll*, *jogging* and *walking2*, the corresponding values decrease from 22.4, 90.7 and 44.9 to 3.1, 2.9 and 4.0, respectively. The results indicate that the proposed algorithm can remarkably improve the tracking performance when integrated with the visual tracking algorithm. The additional benefits from the integration of Mean-shift tracking and foreground segmentation is the scale adaption, which is of great importance for visual tracking.

E. ROBUSTNESS, FAULTINESS AND RUNTIME ANALYSIS

In this subsection, we will mainly analyze the robustness, faultiness and runtime of the proposed algorithm.

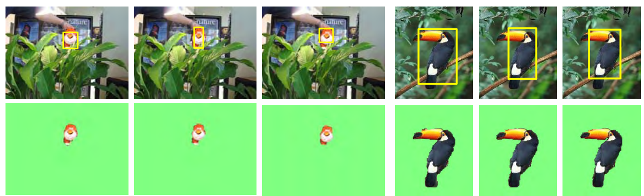


FIGURE 10. Robustness test to different initial bounding-box inputs. For the top row, the left three are *tiger* images with different predefined bounding-box, and the right three are *bird* images with different predefined bounding-box. The bottom row displays the segmentation results.

1) ROBUSTNESS ANALYSIS

The proposed algorithm is robust to the input bounding-box. We evaluate this performance by shifting the input and changing the size and aspect ratio. The segmentation results are robust to different input bounding-boxes. We refer to Fig.10 for a simple demonstration. However, if the coverage ratio of input bounding-box to the ground-truth is less than 0.5, the performance may become worse.

The proposed algorithm involves several parameters, however, it is not sensitive to the parameter change. We test the parameter N_1 that control the Watershed superpixels number and display the test results in Tab.2. We preset the SLIC superpixels number at 200 and increases N_1 from 160 to 240 with step of 20. The results show that the final segmentation changes very little when the Watershed superpixels number variants around 200 and the parameter 200 produces relative better results. We conclude the reasons as two aspects. Firstly, the variation of the preset Watershed superpixels number has more influence to the segmentation of the region with rich image details. However, limited change of this segmentation has very little influence to the initial foreground and background labels extraction. Secondly, we exploit the Mean-shift clustering algorithm to learn the initial foreground and background labels during the label initialization. This algorithm has a relative high robustness to the initial input and thus, a slight change of Watershed superpixels number will not bring in large change of final results.

We also test the robustness to the preset SLIC superpixels number (N_2) from 150 to 250 and displayed the results in Tab. 3. We restrict N_1 at 200 and increase the N_2 with step

TABLE 3. The TPR(%) and FPR(%) values under different parameter settings of N_2 with $N_1 = 200$.

test image	$N_2 = 150$		$N_2 = 175$		$N_2 = 200$		$N_2 = 225$		$N_2 = 250$	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
human	87.44	1.20	90.37	0.41	91.62	0.33	89.53	0.61	89.53	0.61
sloth	86.93	1.37	90.16	0.91	93.44	0.36	91.04	1.18	91.04	1.18
boat	92.61	0.86	95.16	0.34	96.91	0.27	93.39	0.80	93.39	0.80
bird	97.75	0.21	98.19	0.14	99.15	0.13	98.77	0.16	98.77	0.16
fish	86.59	0.78	90.24	0.36	92.56	0.11	89.54	0.75	89.54	0.75
ostrich	90.34	0.82	91.69	0.43	94.49	0.18	92.28	0.63	92.28	0.63
face	96.85	0.69	98.10	0.27	99.32	0.08	97.74	0.71	97.74	0.71
skating	86.30	1.47	89.43	0.52	92.07	0.24	88.89	1.03	88.89	1.03
Average	90.60	0.93	92.92	0.42	**94.95**	**0.21**	92.65	0.73	92.65	0.73

TABLE 4. The runtime (in *ms*) comparison of different superpixel extraction methods.

Methods	50×50	100×100	150×150	200×200	250×250
Mean-shift	11.6	42.7	104.2	176.0	275.8
Watershed	2.3	8.8	22.1	38.3	54.6
SLIC	4.1	9.3	19.6	31.7	49.5

of 25. The results seem very consistent when $N_2 = 225$ and $N_2 = 250$. This is because that the actual SLIC superpixels number is an integral square root of the figure no larger than N_2. So, when $N_2 = 225$ and $N_2 = 250$, the actual SLIC superpixels number are both 225 (equals 15^2). The results show that the algorithm obtains the largest TPR and the smallest FPR in most cases. Actually, the final results will keep the same when N_2 variants from 196 (equals 16^2) to 224 (the actual SLIC superpixels number is still 196, which equals 16^2). We also test the bandwidth of Mean-shift clustering from 0.1 to 0.5 with step of 0.01 and found that a value around 0.25 obtains a relative better result. For the whole algorithm in this paper, we keep these three parameters unchanged for all the experiments.

Note that if the background becomes very complex, the performance of the proposed algorithm may decrease distinctively. So, we advise to further improve the performance by integrating the proposed algorithm with some other strategies that can increase the contrast between foreground and background.

2) FAULTINESS ANALYSIS

To be mentioned is that we confronted with some failing examples during the experiments. For the image *human* and *bird* in Fig.6, some fine details are missed in the final results, such as the long staff in the image *human* and the claws in the image *bird*. The intrinsic reason is that the superpixel extraction algorithm (SLIC) can not segment such fine details and thus, this small foregrounds are segmented into background in the superpixel segmentation step. We tried to enlarge the superpixles number to alleviate this problem and obtain an improved result at the cost of time-consuming increasing. However, if the superpixels method can not distinguish the foreground from background, it will directly affect the final segmentation result. A superpixels method with high precision and efficiency may be helpful to resolve this problem.

3) RUNTIME ANALYSIS

The time-consuming of the proposed algorithm mainly consists of three parts: superpixel extraction, initial labels generation and SSG. Among these three, superpixel extraction is affected by the region size of foreground while initial labels generation and SSG are mainly affected by the number of superpixels and the dimension of feature. In order to evaluate the efficiency of the proposed algorithm, we test it with different foreground size and different superpixels number. We also test the proposed algorithm and the referenced algorithm based on the images from BSD database. All these tests are implemented on a PC with Intel 3.3GHz CPU and 4GB ROM.

In order to compare the time-consuming of Mean-shift clustering algorithm, Watershed algorithm and SLIC algorithm, we randomly generate five groups images with the size from 50×50 to 250×250 with step of 50. Each group contains 100 different images. After, we exploit these three methods to extract the superpixel of all the test images and calculate the average runtime per image. The comparison results are displayed in Tab. 4. From the comparison, we can see that the Watershed algorithm and SLIC algorithm are more efficient than Mean-shift clustering algorithm. It only takes millisecond-level for Watershed algorithm and SLIC algorithm to extract the superpixel of image patch with the size of 100×100. It is of great significance for the integrated algorithm to cut down the time-consuming during the preprocessing.

Tab. 5 displays the time-consuming of the proposed algorithm with the increasing of foreground size. We fix the size of the test image as 481×321, and increase the size of the input bounding-box from 20×20 to 100×100 with step of 20. The parameters N_1 and N_2 (the superpixels number of Watershed and SLIC) are kept as 200. We also recorded the runtime of the referenced algorithms in this table to compare the efficiency of them. In order to test the runtime of GrabCut and LooseCut algorithm, we run the segmentation

TABLE 5. Runtime (in seconds) with the increasing of the input bounding-box size.

Methods	20 × 20	40 × 40	60 × 60	80 × 80	100 × 100
Ours	1.96	2.04	2.52	3.23	3.51
GrabCut	2.98	3.11	3.06	3.01	3.03
MSRM	1.92	1.91	2.03	2.17	2.08
LooseCut	4.59	4.47	4.55	4.63	4.39

TABLE 6. Runtime (in seconds) with the increasing of SLIC superpixels number.

Methods	100	150	200	250	300
Ours	1.74	1.99	2.53	3.23	3.85
GrabCut	3.04	2.95	3.01	3.12	3.07
MSRM	1.68	1.87	2.06	2.85	3.69
LooseCut	4.61	4.70	4.59	4.46	4.57

codes ten times with randomly initialized bounding-boxes to obtain an average value. The runtimes of MSRM are also statistical values based on ten times tests. We can see that the proposed algorithm runs very efficient when the size of the input bounding-box is small. The runtime increases when the size of the input bounding-box increases.

The time-consuming of the proposed algorithm with the increasing of SLIC superpixels number are demonstrated in Tab. 6. The size of the test image is fixed as 481×321. In order to analyze the influence to time-consuming with the increasing of SLIC superpixels number under a single variable maintaining other effect factors constant, we select a value neither too big nor too little (60×60) as the size of input bounding-box. The size of the test image is fixed as 481×321, and the size of input bounding-box is fixed as 60×60. SLIC superpixels number increases from 100 to 300 with step of 50. We found that the runtime of the proposed algorithm increases along with the increasing of the SLIC superpixels number. Actually, if the image size and the input bounding-box size are fixed, the runtime of the proposed algorithm mainly correlates to the SLIC superpixels number. The increasing of SLIC superpixels number will increase the processing time when describing the superpixels and grouping these superpixels using SSG algorithm. It is consistent to the experiment results displayed in Tab. 6.

To be pointed out is that the runtime of MSRM only considers the superpixel grouping step because the input is the over-segmented regions using Mean-shift clustering algorithm. So, the Tables show that MSRM is slightly more efficient than the proposed algorithm. It can be concluded that the proposed algorithm is practically as efficient as MSRM and more efficient than GrabCut and LooseCut. Further more, the proposed algorithm is implemented only using Matlab and thus, it can be substantially optimized for speed. Nevertheless, there is a distance to obtain a realtime processing in current state. We will do further research to improve the efficiency of the proposed algorithm.

V. CONCLUSION

We proposed a weakly supervised image segmentation algorithm to extract foreground from a complex background relying only on a roughly predefined bounding-box. The main contribution of this paper are the weakly supervised foreground segmentation framework based on a roughly predefined bounding-box and the synthetic superpixel grouping mechanism. We cluster the Watershed superpixels to obtain the initial foreground and background labels according to the input bounding-box. A synthetic superpixel grouping algorithm is then proposed to segment the foreground based on the SLIC superpixels according to the initialized foreground and background labels. Extensive experiments indicate that the proposed algorithm can obtain a robust segmentation which outperforms the related works in most cases. We also discussed the applications and analyzed the faultiness of the proposed algorithm to comprehensively evaluate its performance. Future work will be centralized on the integration of SSG with some other superpixel methods and the improving of the efficiency.

ACKNOWLEDGMENT

The authors would like to thank Ning *et al.* for making their MSRM code public available.

REFERENCES

[1] R. G. Cinbis, J. Verbeek, and C. Schmid, "Segmentation driven object detection with fisher vectors," presented at the 14th IEEE Int. Conf. Comput Vis., Sydney, NSW, Australia, Dec. 2013, pp. 2968–2975.

[2] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[3] R. Almomani and M. Dong, "SegTrack: A novel tracking system with improved object segmentation," presented at the 20th IEEE Int. Conf. Image Process., Melbourne, VIC, Australia, Sep. 2013, pp. 3939–3943.

[4] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," presented at the 1st IEEE Int. Conf. Comput. Vis., London, U.K., Jun. 1987, pp. 259–268.

[5] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[6] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[7] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," presented at the 8th IEEE Int. Conf. Comput. Vis., Vancouver, BC, Canada, Jul. 2001, pp. 105–112.

[8] O. Juan and Y. Boykov, "Active graph cuts," presented at the 18th IEEE Conf. Comput. Vis. Pattern Recognit., New York, NY, USA, Jun. 2006, pp. 1023–1029.

[9] C. Rother, V. Kolmogorov, and A. Blake, "grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[10] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognit.*, vol. 43, no. 2, pp. 445–456, Feb. 2010.

[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. 2274–2281, Nov. 2012.

[12] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[13] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, ''SEEDS: Superpixels extracted via energy-driven sampling,'' *Int. J. Comput. Vis.*, vol. 111, no. 3, pp. 298–314, 2015.

[14] H. W. Kang and S. Y. Shin, ''Enhanced lane: Interactive image segmentation by incremental path map construction,'' *Graph. Models.*, vol. 64, no. 5, pp. 282–303, Sep. 2002.

[15] Y. Li, J. Sun, C. Tang, and H. Shum, ''Lazy snapping,'' *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, Aug. 2004.

[16] T. Wang, B. Han, and J. Collomosse, ''TouchCut: Fast image and video segmentation using single-touch interaction,'' *Comput. Vis. Image Understand.*, vol. 120, pp. 14–30, Mar. 2014.

[17] M. S. Farid, A. Mahmood, and M. Grangetto, ''Image de-fencing framework with hybrid inpainting algorithm,'' *Signal Image Video Process.*, vol. 10, no. 7, pp. 1193–1201, Oct. 2016.

[18] M. S. Farid, M. Lucenteforte, M. H. Khan, and M. Grangetto, ''Semi-automatic segmentation of scattered and distributed objects,'' presented at 10th Int. Conf. Comput. Recognit. Syst., May 2017, pp. 110–119.

[19] J. Shen, Y. Du, and X. Li, ''Interactive segmentation using constrained Laplacian optimization,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1088–1100, Jul. 2014.

[20] J. Shen, Y. Du, W. Wang, and X. Li, ''Lazy random walks for superpixel segmentation,'' *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.

[21] X. Dong, J. Shen, L. Shao, and L. Van Gool, ''Sub-Markov random walk for image segmentation,'' *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 516–527, Feb. 2016.

[22] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, ''Real-time superpixel segmentation by DBSCAN clustering algorithm,'' *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5933–5942, Dec. 2016.

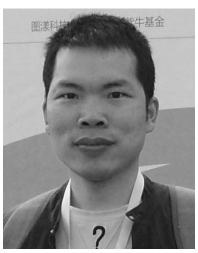
[23] H. Yu. (2015). ''LooseCut: Interactive image segmentation with loosely bounded boxes,'' Accessed: Jul. 11, 2015. [Online]. Available: https://arxiv.org/abs/1507.03060

[24] C. Y. Ren, V. A. Prisacariu, and I. D. Reid. ''gSLICr: SLIC superpixels at over 250 Hz,'' Accessed: Sep. 14, 2015. [Online]. Available: https://arxiv.org/abs/1509.04232

[25] W. Yu, X. Tian, Z. Hou, and Y. Zha, ''Robust visual tracking based on watershed regions,'' *IET Comput. Vis.*, vol. 8, no. 6, pp. 588–600, 2014.

[26] D. Martin, C. Fowlkes, D. Tal, and J. Malik, ''A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,'' presented at the 8th IEEE Int. Conf. Comput Vis., Vancouver, BC, Canada, Jul. 2001, pp. 416–423.

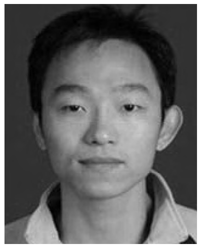
[27] Y. Wu, J. Lim, and M. H. Yang, ''Object tracking benchmark,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.



WANGSHENG YU was born in Hunan, China, in 1985. He received the B.S. degree in command automation system and the M.S. and Ph.D. degrees in communication and information system from Air Force Engineering University (AFEU) in 2008, 2010, and 2014, respectively.

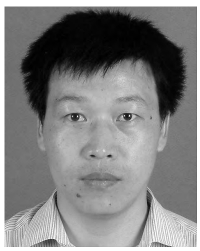
Since 2014, he has been a Lecturer with the Information and Navigation College, AFEU. He has published over 80 articles. His current research interests include computer vision, image processing, and deep learning.

ZHIQIANG HOU was born in Shaanxi, China. He received the M.S. degree from Air Force Engineering University in 1998 and the Ph.D. degree from Xian Jiaotong University in 2005. He was a Visiting Scholar with the University of Bristol, U.K., in 2009. He is currently a Professor with the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications. His research interests include pattern recognition, computer vision, image processing, and information fusion.

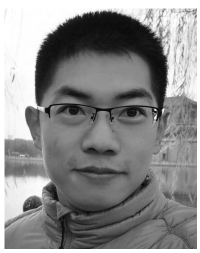


PENG WANG was born in Shanxi, China. He received the B.S. and M.S. degrees from Chang'an University, Xi'an, China, in 2006 and 2009, respectively, and the Ph.D. degree from Xi'an Jiaotong University in 2013. He is currently a Lecturer with Air Force Engineering University. His research interests include the cooperation of multi-agent systems, distributed estimation, and control based on computer vision.

XIANXIANG QIN received the B.S. degree in electronic and information engineering from Guangxi University, Nanning, China, in 2008, the M.S. degree in information engineering from the National University of Defense Technology (NUDT) in 2010, and the Ph.D. degree in circuits and system from NUDT in 2015. He is currently a Lecturer with the Information and Navigation College, Air Force Engineering University, Xi'an, China. His current research interests include the statistical modeling, target detection, and scene understanding of synthetic aperture radar images.



LIGUANG WANG was born in Shaanxi, China. He received the M.S. degree from Air Force Engineering University in 2006 and the Ph.D. degree from Northwestern Polytechnical University in 2009. He is currently a Lecturer with the Information and Navigation College, Air Force Engineering University. His research interests include the information fusion and target tracking.



HUANYU LI was born in Heilongjiang, China. He received the M.S. and Ph.D. degrees in computer application technology from Air Force Engineering University in 2009 and 2012, respectively. He is currently a Lecturer with the Air Traffic Control and Navigation College, Air Force Engineering University. His research interests include machine learning and pattern recognition.