LooseCut: Interactive Image Segmentation with Loosely Bounded Boxes

Hongkai Yu[†], Youjie Zhou[†], Hui Qian[‡], Min Xian[*], Yuewei Lin[†], Dazhou Guo[†], Kang Zheng[†], Kareem Abdelfatah[†] and Song Wang[†]

[†]Department of Computer Science & Engineering, University of South Carolina, Columbia, SC 29208
[‡]College of Computer Science, Zhejiang University, Hangzhou, China
[*]Department of Computer Science, Utah State University, Logan, UT 84341
{yu55, zhou42}@email.sc.edu, qianhui@zju.edu.cn, min.xian@aggiemail.usu.edu,
{lin59, guo22, zheng37}@email.sc.edu, ker00@fayoum.edu.eg and songwang@cec.sc.edu

Abstract

One popular approach to interactively segment the foreground object of interest from an image is to annotate a bounding box that covers the foreground object. Then, a binary labeling is performed to achieve a refined segmentation. One major issue of the existing algorithms for such interactive image segmentation is their preference of an input bounding box that tightly encloses the foreground object. This increases the annotation burden, and prevents these algorithms from utilizing automatically detected bounding boxes. In this paper, we develop a new LooseCut algorithm that can handle cases where the input bounding box only loosely covers the foreground object. We propose a new Markov Random Fields (MRF) model for segmentation with loosely bounded boxes, including a global similarity constraint to better distinguish the foreground and background, and an additional energy term to encourage consistent labeling of similar-appearance pixels. This MRF model is then solved by an iterated max-flow algorithm. In the experiments, we evaluate LooseCut in three publicly-available image datasets, and compare its performance against several state-of-the-art interactive image segmentation algorithms. We also show that LooseCut can be used for enhancing the performance of unsupervised video segmentation and image saliency detection.

1. Introduction

Accurately segmenting a foreground object of interest from an image with convenient human interactions plays a central role in image and video editing. One widely used interaction is to annotate a bounding box around the foreground object. On one hand, this input bounding box provides the spatial location of the foreground. On the other hand, based on the image information within and outside this bounding box, we can have an initial estimation of the appearance models of the foreground and background, with which a binary labeling is finally performed to achieve a refined segmentation of the foreground and background [15, 17, 16, 18, 13, 10].
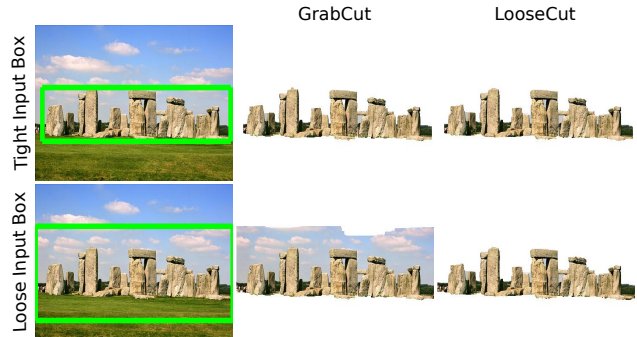

Figure 1. Sample results from GrabCut and the proposed LooseCut with tightly and loosely bounded boxes.

However, due to the complexity of the object boundary and appearance, most of the existing methods of this kind prefer the input bounding box to tightly enclose the foreground object. An example is shown in Fig. 1, where the widely used GrabCut [15] algorithm fails when the bounding box does not tightly cover the foreground object. The preference of a tight bounding box increases the burden of the human interaction, and moreover it prevents these algorithms from utilizing automatically generated bounding boxes, such as boxes from object proposals [2, 23, 22], that are usually not guaranteed to tightly cover the foreground object. In this paper, we focus on developing a new LooseCut algorithm that can accurately segment the foreground object with loosely-bounded boxes.

A loosely bounded box may contain more background than a tightly bounded box. As a result, the initial ap-

pearance model of the foreground is highly inaccurate by using the pixels within the bounding box. This may substantially reduce the segmentation performance as shown by the Grabcut result in Fig. 1. In this paper, we propose two strategies to address this problem. First, we explicitly emphasize the appearance difference between the foreground and background models. Second, we explicitly encourage the consistent labeling to the similar-appearance pixels, either adjacent or non-adjacent. These two strategies can help identify the background pixels within the bounding box, as shown in Fig. 2.
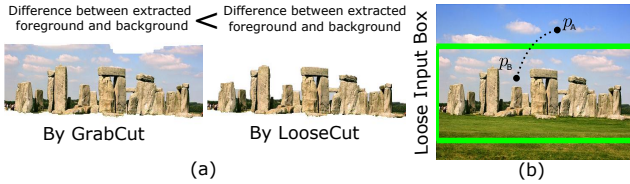

Figure 2. An illustration of the two strategies used in the proposed LooseCut algorithm. (a) By emphasizing their appearance difference, foreground and background are better separated even with a loosely bounding box. (b) By encouraging label consistency of similar-appearance pixels, background pixel P_B inside the loosely bounded box is correctly labeled as background due to its appearance similarity to the background pixel P_A outside the bounding box.

In this paper, we follow GrabCut by formulating the foreground/background segmentation as a binary labeling over an MRF built upon the image grid, and the appearances of the foreground and background are described by two Gaussian Mixture Models (GMMs). More specifically, we add a *global similarity constraint* and a *label consistency term* to the MRF energy to implement the above mentioned two strategies. Finally, we solve the proposed MRF model using an iterated max-flow algorithm. In the experiments, we evaluate the proposed LooseCut in three publicly-available image datasets, and compare its performance against several state-of-the-art interactive image segmentation algorithms. We also show that LooseCut can be used for enhancing the performance of unsupervised video segmentation and image saliency detection.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes the proposed LooseCut algorithm in detail. Section 4 reports the experimental results, followed by a briefly conclusion in Section 5.

2. Related Work

In recent years, interactive image segmentation based on input bounding boxes have drawn much attention in the computer vision and graphics community, resulting in a number of effective algorithms [15, 17, 16, 18, 13, 10]. Starting from the classical GrabCut algorithm, many of these algorithms use graph cut models: the input image is modeled by a graph and the foreground/background segmentation is then modeled by a binary graph cut that minimizes a pre-defined energy function [6]. In GrabCut [15], initial appearance models of the foreground and background are estimated using the image information within and outside the bounding box. A binary MRF model is then applied to label each pixel as the foreground or background, based on which the appearance models of the foreground and background are re-estimated. This process is repeated until convergence. As illustrated in Fig. 1, the performance of GrabCut is highly dependent on the initial estimation of the appearance models of the foreground and background, which might be very poor when the input bounding box does not tightly cover the foreground object. The LooseCut algorithm developed in this paper also follows the general procedure introduced in GrabCut, but introduce a new constraint and a new energy term to the MRF model to specifically handle the loosely-bounded boxes.

PinPoint [13] is another MRF-based algorithm for interactive image segmentation with a bounding box. It incorporates a topology prior derived from geometry properties of the bounding box and encourages the segmented foreground to be tightly enclosed by the bounding box. Therefore, its performance gets much worse with a loosely bounded box. Also using an MRF model, OneCut [17] is recently developed for interactive image segmentation. Its main contribution is to incorporate an MRF energy term that reflects the appearance overlap between foreground and background histograms. As shown in the latter experiments, the L_1-distance based appearance overlap used in OneCut is still insufficient to handle loosely-bounded boxes. In [16], a pPBC algorithm is developed for interactive image segmentation using an efficient parametric pseudo-bound optimization strategy. However, in our experiment shown in Section 4, pPBC still cannot give satisfactory segmentation results when the input bounding box is loose.

Other than using the MRF model, MILCut [18] formulates the interactive image segmentation as a multiple instance learning problem by generating positive bags along the sweeping lines within the bounding box. MILCut may not generate the desirable positive bags along the sweeping lines for a loosely bounded box. Active contour [10] takes the input bounding box as an initial contour and iteratively deforms it toward the boundary of the foreground object. Due to its sensitivity to image noise, active contour usually requires the initial contour to be close to the underlying foreground object boundary.

3. Proposed Approach

In this section, we first briefly review the classical GrabCut algorithm and then explain the proposed LooseCut algorithm.

3.1. GrabCut

GrabCut [15] actually performs a binary labeling to each pixel using an MRF model. Let $X = \{x_i\}_{i=1}^n$ be the binary labels at each pixel i, where $x_i = 1$ if i is in foreground $x_i = 0$ if i is in background and let $\theta = (M_f, M_b)$ denotes the appearance models including foreground GMM M_f and background GMM M_b. Grabcut seeks an optimal labeling that minimizes

$$E_{GC}(X, \theta) = \sum_i D(x_i, \theta) + \sum_{i,j \in \mathcal{N}} V(x_i, x_j), \quad (1)$$

where $\mathcal{N}$ defines a pixel neighboring system, e.g., 4-neighbor or 8-neighbor connectivity. The unary term $D(x_i, \theta)$ measures the cost of labeling pixel i as foreground or background based on the appearance models θ. The pairwise term $V(x_i, x_j)$ enables the smoothness of the labels by penalizing discontinuity among the neighboring pixels with different labels. Max-flow algorithm [6] is usually used for solving this MRF optimization problem. GrabCut takes the following steps to achieve the binary image segmentation with an input bounding box:

1. Estimating initial appearance models θ, using the pixels inside and outside the bounding box respectively.

2. Based on the current appearance models θ, quantizing the foreground and background likelihood of each pixel and using it to define the unary term $D(x_i, \theta)$. And solve for the optimal labeling that minimizes Eq. (1).

3. Based on the obtained labeling X, refining θ and going back to Step 2. Repeating this process until convergence.

3.2. MRF Model for LooseCut

Following the MRF model used in GrabCut, the proposed LooseCut takes the following MRF energy function:

$$E(X, \theta) = E_{GC}(X, \theta) + \beta E_{LC}(X), \quad (2)$$

where E_{GC} is the GrabCut energy given in Eq. (1), and E_{LC} is an energy term for encouraging label consistency, weighted by $\beta > 0$. In minimizing Eq. (2), we enforce a global similarity constraint to better estimate θ and distinguish the foreground and background. In the following, we elaborate on the global similarity constraint and the label consistency term $E_{LC}(X)$.

3.3. Global Similarity Constraint

In this section, we define the proposed global similarity constraint. Let M_f have K_f Gaussian components M_f^i with means μ_f^i, $i = 1, 2, \cdots, K_f$ and M_b have K_b Gaussian components M_b^j with means μ_b^j, $j = 1, 2, \cdots, K_b$. For each Gaussian component M_f^i in the foreground GMM M_f, we first find its nearest Gaussian component $M_b^{j(i)}$ in M_b as

$$j(i) = \arg \min_{j \in \{1, \ldots, K_b\}} \left| \mu_f^i - \mu_b^j \right|. \quad (3)$$

With this, we can define the similarity between the Gaussian component M_f^i and the entire background GMM M_b as

$$S\left(M_f^i, M_b\right) = \frac{1}{\left| \mu_f^i - \mu_b^{j(i)} \right|}, \quad (4)$$

which is the inverse of the mean difference between M_f^i and its nearest Gaussian component in the background GMM. Then, we define the global similarity function Sim as

$$Sim(M_f, M_b) = \sum_{i=1}^{K_f} S\left(M_f^i, M_b\right). \quad (5)$$

Similar definition for GMM distance could be found in [19]. In the MRF minimization, we will enforce the global similarity $Sim(M_f, M_b)$ to be low (smaller than a threshold) in the step of estimating θ and details will be discussed in Section 3.5.

3.4. Label Consistency Term E_{LC}

To encourage the label consistency of the similar-appearance pixels, either adjacent or non-adjacent, we first cluster all the image pixels using a recent superpixel algorithm [21] that preserves both feature and spatial consistency. Following a K-means-style procedure, this cluster algorithm partitions the image into a set of compact superpixels and each resulting cluster is made up of one or more superpixels. An example is shown in Fig. 3, where the region color indicates the clusters: superpixels with the same color constitute a cluster.
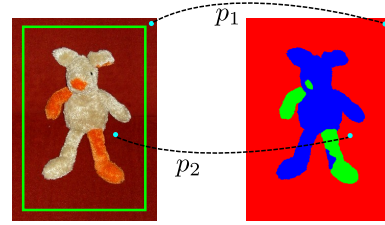

Figure 3. An illustration of the superpixel based clusters and label consistency. Three clusters are shown in different colors.

Let C_k indicates the cluster k, and pixels belonging to C_k should be encouraged to be given the same label, e.g., p_1 and p_2 in Fig. 3. To accomplish this, we set a cluster label x_{C_k} (taking values 0 or 1) for each cluster C_k and define the label-consistency energy term as

$$E_{LC}(\mathbf{X}) = \sum_k \sum_{i \in C_k} \phi(x_i \neq x_{C_k}), \quad (6)$$

where $\phi(\cdot)$ is an indicator function taking 1 or 0 for true or false argument. In the proposed algorithm, we will solve for

3

both the pixel labels and cluster labels simultaneously in the MRF optimization.

3.5. Optimization

In this section, we propose an algorithm to find the optimal binary labeling that minimizes the energy function defined in Eq. (2), subject to the global similarity constraint. Specifically, in each iteration, we first fix the labeling X and optimize over θ by enforcing the global similarity constraint on $Sim(M_f, M_b)$. After that, we fix θ and find an optimal X that minimizes $E(X, \theta)$. These two steps of optimization is repeated alternately until convergence or a preset maximum number of iterations is reached. As an initialization, we use the input bounding box to define a binary labeling X in iteration 0. In the following, we elaborate on these two optimization steps.

Fixing X and Optimizing over θ: With fixed binary labeling X, we can estimate θ using a standard EM-based clustering algorithm: All the pixels with label 1 are taken for computing the foreground GMM M_f and all the pixels with label 0 are used for computing the background GMM M_b. We intentionally select K_f and K_b such that $K = K_f - K_b > 0$ since some background components are mixed to the foreground for the initial X defined by a loosely bounded box. For the obtained M_f and M_b, we examine whether the global similarity constraint is satisfied, i.e, $Sim(M_f, M_b) \leq \delta$ or not. If this constraint is satisfied, we take the resulting θ and continue to the next step of optimization. If this constraint is not satisfied, we further refine M_f using the following algorithm:

1. Calculate the similarity $S(M_f^i, M_b)$ between each Gaussian component of M_f and M_b, by following Eq. (4) and identify the K Gaussian components of M_f with the largest similarity to M_b.

2. Among these K components, if any one, say M_f^i, does not satisfy $S(M_f^i, M_b) \leq \delta$, we delete it from M_f.

3. After all the deletions, we use the remaining Gaussian components to construct an updated M_f.

This algorithm will ensure the updated M_f and M_b satisfies the global similarity constraint.

Fixing θ and Optimizing over X: Inspired by [11] and [17], we build an undirect graph with auxiliary nodes as shown in Fig. 4 to find an optimal X that minimizes the energy $E(X, \theta)$. In this graph, each pixel is represented by a node. For each pixel cluster C_k, we construct an auxiliary node A_k to represent it. Edges are constructed to link the auxiliary node A_k and the nodes that represent the pixels in C_k, with the edge weight β as used in Eq. (2). An example of the constructed graph is shown in Fig. 4, where pink nodes v_1, v_5, and v_6 represent three pixels in a same cluster, which is represented by the auxiliary node A_1. All

the nodes in blue represent another cluster. With a fixed θ, we use the max-flow algorithm [6] on this graph to seek an optimal X that minimizes the energy $E(X, \theta)$.
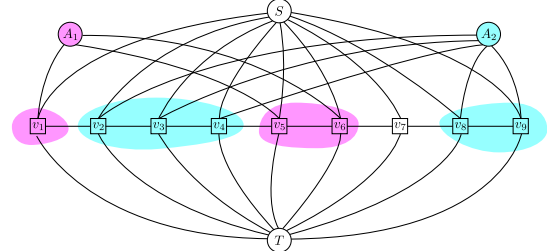

Figure 4. Graph construction for the step of optimizing over X with a fixed θ. v_i's are the nodes for pixels and A_i's are the auxiliary nodes. S and T are the source and sink nodes. Same color nodes represent a cluster.

The graph constructed as in Fig. 4 is similar to the graph constructed in OneCut [17]. However, there are two major differences between the proposed algorithm and OneCut.

1. In OneCut, a color histogram is first constructed for the input image and then one auxiliary node is constructed for each histogram bin. All the pixels are then quantized into these bins and the pixels in each bin are then linked to its corresponding auxiliary node. In this paper, we use superpixel-based clusters to define the auxiliary nodes.

2. The unary energy term in OneCut is different from the one in the proposed method and as a result, we define the edge weights involving the source and sink nodes differently from OneCut. OneCut follows the ballooning technique: The weight is set to 1 for the edges between S and any pixels inside the bounding box, and 0 otherwise; Similarly, the weight is set to 0 for the edges between T and any pixels in the bounding box, and ∞ otherwise. In the proposed algorithm, the weights of the edges that are incident from S or T reflect the unary term in Eq. (2), which is based on the appearance models θ.

With these two differences, OneCut seeks to minimize the L_1-distance based histogram overlap between the foreground and background. This is different from the goal of the proposed algorithm: we seek better label consistency of the pixels in the same cluster by using this graph structure. We will compare with OneCut in the latter experiments. The full LooseCut algorithm is summarized in Algorithm 1.

4. Experiments

To justify the proposed LooseCut algorithm, we conduct experiments on three widely used image datasets – the GrabCut dataset [15], the Weizmann dataset [3, 5], and the iCoseg dataset [4], and compare its performance

4

Algorithm 1 LooseCut

Input: Image I, bounding box B, # of clusters N
Output: Binary labeling X to pixels in I

1: Construct N superpixel based clusters using [21].
2: Create initial labeling X using box B.
3: **repeat**
4:     Based on the current labeling X, estimate and update θ by enforcing $Sim(M_f, M_b) \leq \delta$.
5:     Construct the graph using the updated θ with N auxiliary nodes as shown in Fig. 4.
6:     Apply the max-flow algorithm [6] to update labeling X by minimizing $E(X, \theta)$.
7: **until** Convergence or maximum iterations reached

against several state-of-the-art interactive image segmentation methods, including GrabCut [15], OneCut [17], MIL-Cut [18], and pPBC [16]. We also conduct experiments to show the effectiveness of LooseCut in two applications: unsupervised video segmentation and image saliency detection.

Metrics: As in [18] [17] [13], we use *Error Rate* to evaluate an interactive image segmentation by counting the percentage of misclassified pixels inside the bounding box. We also take the pixel-wise *F-measure* as an evaluation metric, which combines the precision and recall metrics in terms of the ground-truth segmentation.

Parameter Settings: For the number of Gaussian components in GMMs, K_b is set to 5 and K_f is set to 6. As discussed in Section 3.5, $K = K_f - K_b = 1$. To enforce the global similarity constraint, we delete $K = 1$ component in M_f. The number of clusters (auxiliary nodes in graph) is set to $N = 16$. For the LooseCut energy defined in Eq. (2), we consistently set $\beta = 0.01$. The unary term and binary term in Eq. (2) are the same as in [15] and RGB color features are used to construct the GMMs. We set $\delta = 0.02$ in deleting the foreground GMM component to enforce the global similarity constraint. For all the comparison methods, we follow their default or recommended settings in their codes.

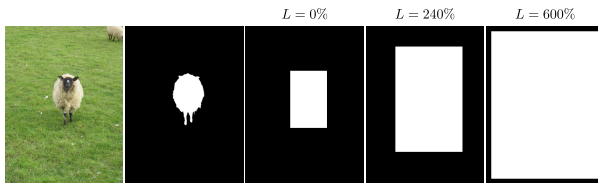$$L = 0\% \qquad L = 240\% \qquad L = 600\%$$

Figure 5. Bounding boxes with different looseness. From left to right: image, ground-truth foreground, baseline bounding box and a series of bounding boxes with increased looseness.

4.1. Interactive Image Segmentation

In this experiment, we construct bounding boxes with different looseness and examine the resulting segmentation. As illustrated in Fig. 5, we compute the fit box to the ground-truth foreground and slightly dilate it by 10 pixels along four directions, i.e., left, right, up, and down. We take it as the baseline bounding box with 0% looseness. We then keep dilating this bounding box uniformly along all four directions to generate a series of looser bounding boxes – a box with a looseness L (in percentage) indicates its area increase by L against the baseline bounding box. A bounding box will be cropped when any of its sides reaches the image perimeter. An example is shown in Fig. 5.

GrabCut dataset [15] consists of 50 images. Nine of them contain multiple objects while the ground truth is only annotated on a single object, e.g., ground truth only label one person but there are two people in the loosely bounded box. Such images are not applicable to test performance change when we enlarge the box looseness. Therefore, we use the remaining 41 images in our experiments. From Weizmann dataset [3, 5], we pick a subset of 45 images for testing, by discarding the images where the baseline bounding box has almost cover the full image and cannot be dilated to construct looser bounding boxes. For the similar reason, from iCoseg dataset [4], we select a subset of 45 images for our experiment.

Experimental results on these three datasets are summarized in Fig. 6. In general, the segmentation performance degrades when the bounding-box looseness increases for both the proposed LooseCut and all the comparison methods. However, LooseCut shows a slower performance degradation than the comparison methods. When the looseness is high, e.g., $L = 300\%$ or $L = 600\%$, LooseCut shows much higher F-measure and much lower Error Rate than all the comparison methods. Since MILCut's code is not publicly available, we only report MILCut's F-measure and Error Rate values with the baseline bounding boxes on the GrabCut dataset and the Weizmann dataset by copying it from the original paper. Table 1 reports the values of F-measure and Error Rate of segmentation with varying-looseness bounding boxes on GrabCut dataset. Sample segmentation results, together with the input bounding boxes with different looseness, are shown in Fig. 4.

4.2. Unsupervised Video Segmentation

The goal of unsupervised video segmentation is to automatically segment the objects of interest from each video frame. The segmented objects can then be associated across frames to infer the motion and action of these objects. It is important for video analysis and semantic understanding [8]. One popular approach for unsupervised video segmentation is to detect a set of object proposals, in the form of bounding boxes [12], from each frame and then extract the
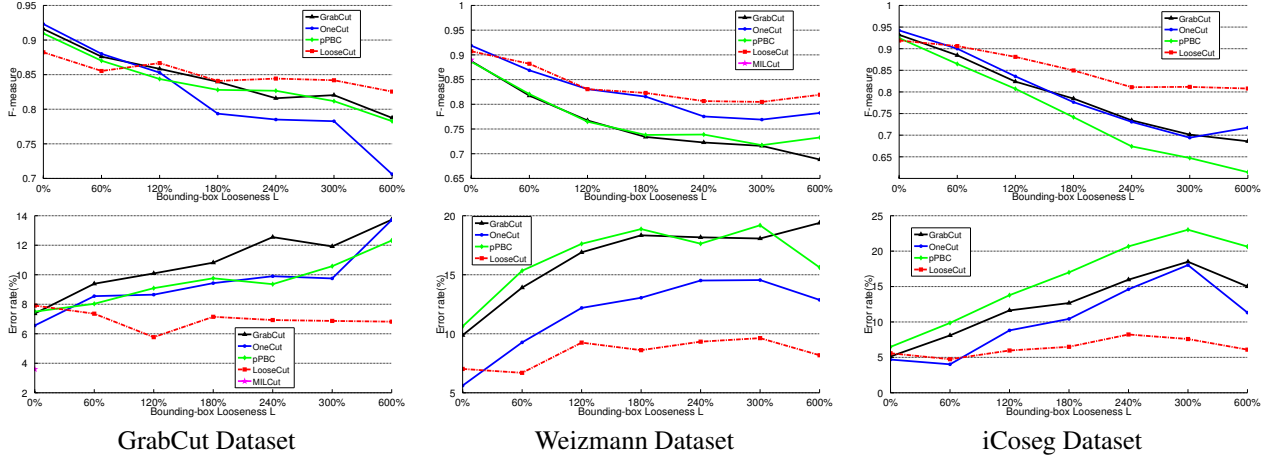
GrabCut Dataset   Weizmann Dataset   iCoseg Dataset

Figure 6. Interactive image segmentation performance (top: F-measure; bottom: Error Rate) on three widely used datasets.

Methods	$L = 0\%$		$L = 120\%$		$L = 240\%$		$L = 600\%$	
	F-measure	Error Rate	F-measure	Error Rate	F-measure	Error Rate	F-measure	Error Rate
GrabCut	0.916	7.4	0.858	10.1	0.816	12.6	0.788	13.7
OneCut	**0.923**	6.6	0.853	8.7	0.785	9.9	0.706	13.7
pPBC	0.910	7.5	0.844	9.1	0.827	9.4	0.783	12.3
MILCut	-	**3.6**	-	-	-	-	-	-
LooseCut	0.882	7.9	**0.867**	**5.8**	**0.844**	**6.9**	**0.826**	**6.8**

Table 1. Segmentation performance on GrabCut dataset with bounding boxes of different looseness.

objects of interest from these proposals [20].

In practice, a detected proposal may only cover part of the object of interest, so we detect a set of object proposals and merge them together to construct a large mask, which has a better chance to cover the whole object. Clearly, this merged mask may only loosely bound the object of interest and the object could be extracted by mask based segmentation algorithms. Specifically, we apply a recent FusionEdgeBox algorithm [22] to detect top 10 object proposals in each video frame for the merged mask.

This experiment is conducted on a subset (21 videos, 657 frames) of JHMDB video dataset [9]. Table 2 shows the unsupervised video segmentation performance, in terms of F-measure and Error Rate averaged over all the frames. We can see that the proposed LooseCut substantially outperforms GrabCut, OneCut and pPBC in this task. Sample video segmentation results are shown in Fig. 8.

Methods	F-measure	Error Rate
FusionEdgeBox Mask	0.35	77.0
GrabCut	0.55	30.5
OneCut	0.58	25.1
pPBC	0.54	31.6
LooseCut	**0.64**	**17.0**

Table 2. Unsupervised video segmentation performance.

4.3. Image Saliency Detection

Recently, GrabCut has been used to detect the salient area from an image [14]. As illustrated in Fig. 9: a set of pre-defined bounding boxes are overlaid to the input image and with each bounding box, GrabCut is applied for a foreground segmentation. The probabilistic saliency map is finally constructed by combining all the foreground segmentations. In this experiment, it is clear that many pre-defined bounding boxes are not tight.

In this experiment, out of 1000 images in the Salient Object dataset [1], we randomly select 100 images for testing. 15 pre-defined masks are shown in Fig. 9. For quantitative evaluation, we follow [1] to binarize a resulting saliency map using an adaptive threshold (two times the mean saliency of the map). Table 3 reports the precision, recall and F-measure of saliency detection when using GrabCut, OneCut, pPBC, and LooseCut for foreground segmentation. We also include comparisons of two state-of-the-art saliency detection methods that do not use pre-defined masks, namely FT [1] and RC [7]. Sample saliency detection results are shown in Fig. 10.

We can see that LooseCut outperforms GrabCut, OneCut and pPBC in this task. It also outperforms FT which does not use bounding-box based segmentation. RC [7] achieves the best performance for saliency detection, because it combines more complex saliency cues than segmentation based approach.
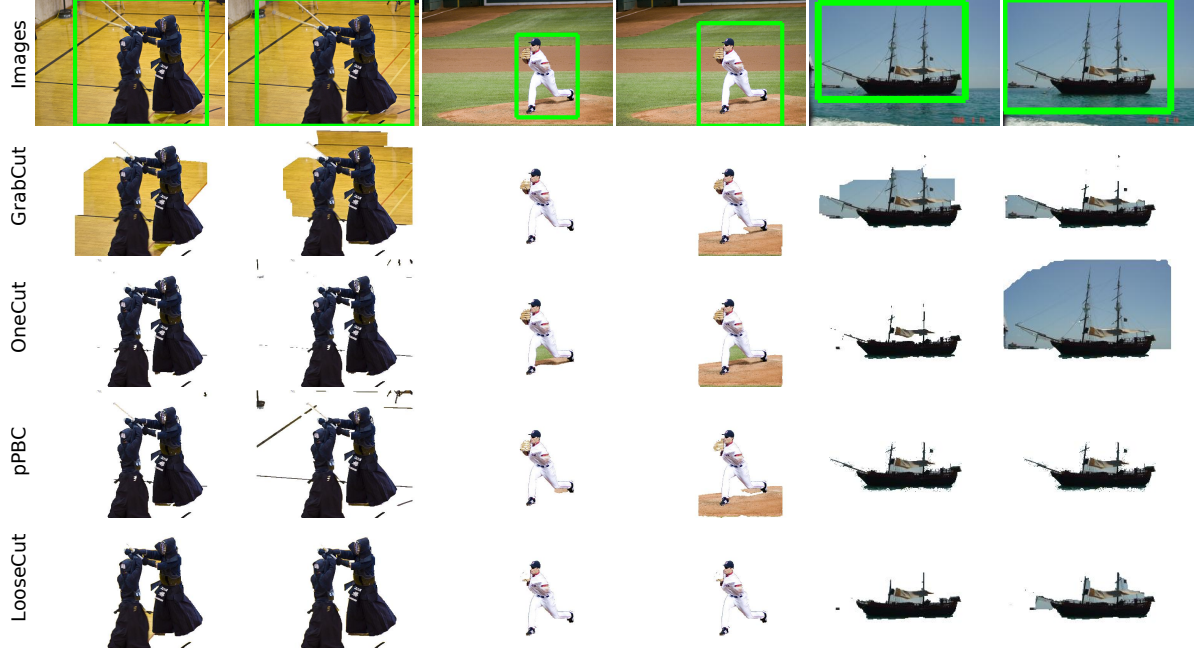
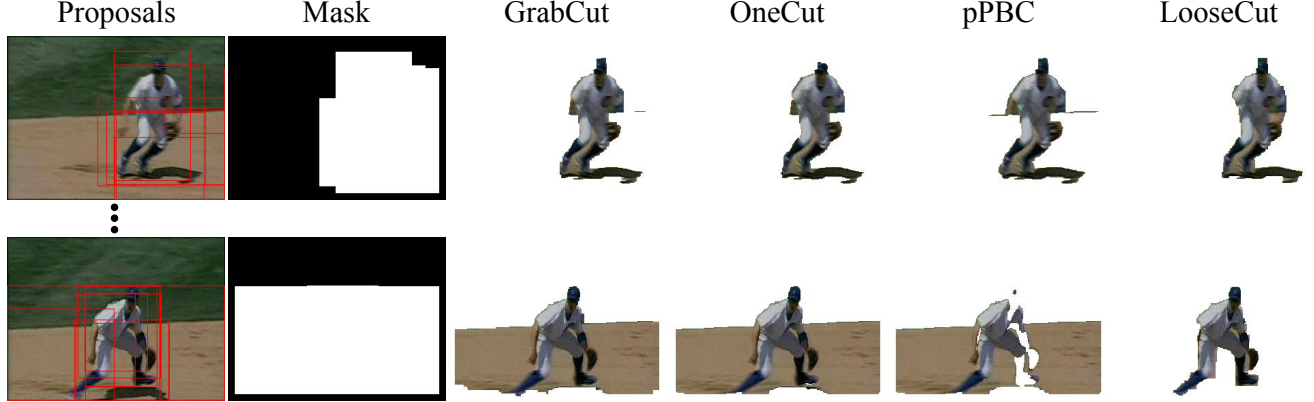Figure 7. Sample results for interactive image segmentation.

Figure 8. Sample video segmentation. From left to right: top 10 detected object proposals (red rectangles), merged mask and different segmentation results.
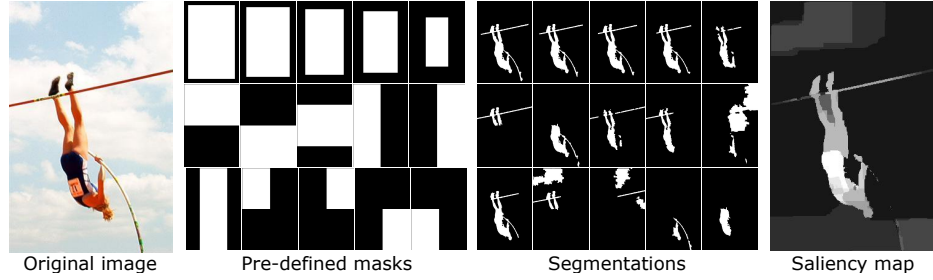

Figure 9. Segmentation based saliency detection.

Methods	GrabCut Dataset		Weizman Dataset		iCoseg Dataset	
	F-measure	Error Rate	F-measure	Error Rate	F-measure	Error Rate
LooseCut w/o proposed constraint & term	0.788	13.7	0.688	19.4	0.686	15.0
LooseCut w/o global similarity constraint	0.801	12.0	0.709	17.9	0.691	14.8
LooseCut w/o label consistency term	0.822	7.3	0.836	7.4	0.806	6.3
LooseCut	**0.826**	**6.8**	**0.841**	**6.6**	**0.808**	**6.1**

Table 4. The usefulness of the proposed global similarity constraint and the label consistency term in LooseCut.

7

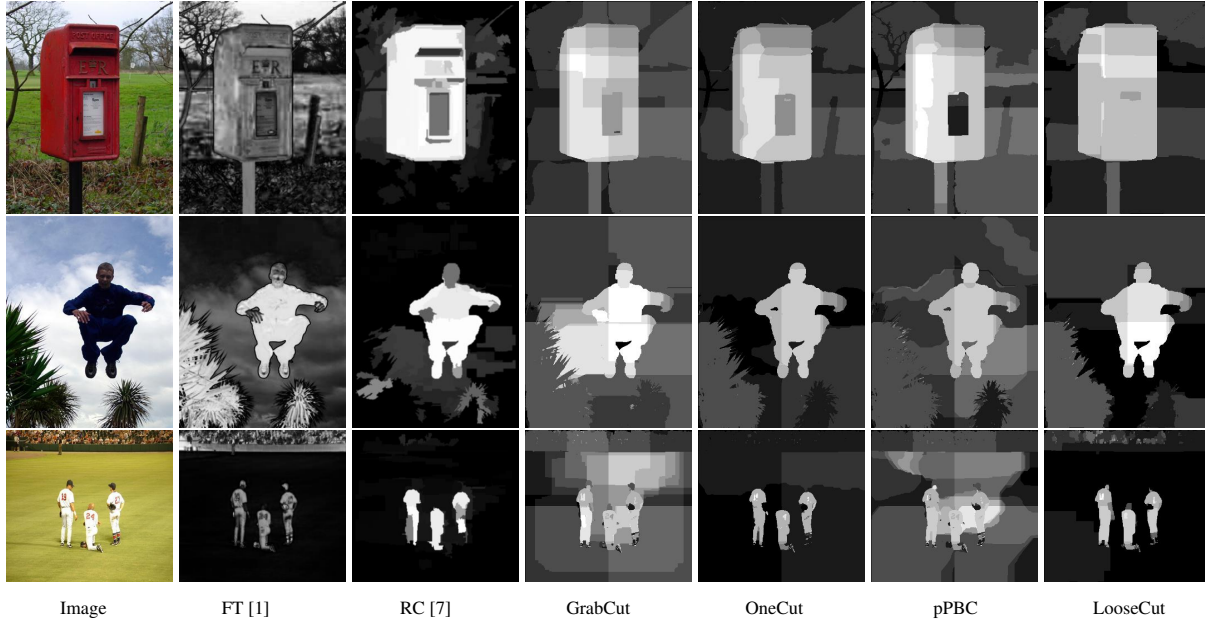| Image | FT [1] | RC [7] | GrabCut | OneCut | pPBC | LooseCut |

Figure 10. Sample saliency detection results.

Methods	Precision	Recall	F-measure
FT [1]	0.75	0.57	0.61
RC [7]	0.86	0.85	0.84
GrabCut	0.85	0.61	0.67
OneCut	**0.86**	0.76	0.77
pPBC	0.84	0.66	0.69
LooseCut	0.84	**0.78**	**0.78**

Table 3. Performance of saliency detection.

4.4. Additional Results

In this section, we report additional results that justify the usefulness of the global similarity constraint and the label consistency term, the running time of the proposed algorithm and possible failure cases.

We run experiments on the three image segmentation datasets when $L = 600\%$ by removing the global similarity constraint and/or the label consistency term, together with their corresponding optimization steps in the proposed LooseCut algorithm. The quantitative performance is shown in Table 4. We can see that both the global similarity constraint and the label consistency term help improve the segmentation performance. The global similarity constraint helps improve the segmentation performance more significantly than the label consistency term.

For the running time, we test LooseCut and all the comparison methods on a PC with Intel 3.3GHz CPU and 4GB RAM. We compares their running time for different image size. In this experiment, OneCut only has one iteration, and the iterations of GrabCut and LooseCut are stopped until

convergence or a maximum 10 iterations is reached. As shown in Table 5, if the image size is less than 512×512, the running time of three algorithms are very close. For large images, LooseCut and OneCut takes more time than GrabCut. In general, LooseCut still shows reasonable running time. Our current LooseCut code is implemented in Matlab and C++, and it can be substantially optimized for speed.

Methods	64*64	128*128	256*256	512*512	1024*1024
GrabCut	0.16	0.28	1.47	3.81	25.21
OneCut	0.03	0.09	0.49	5.72	77.80
pPBC	0.14	0.37	2.70	26.14	305.60
LooseCut	0.32	0.43	1.68	7.63	66.52

Table 5. Running time (in seconds) with increasing image size.

Due to the proposed global similarity constraint and label consistency term, LooseCut may fail when the foreground and background show highly similar appearances, as shown in Fig. 11.
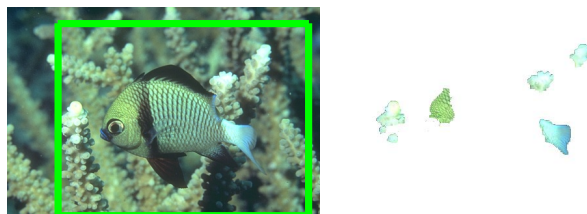

Figure 11. Failure cases of LooseCut.

8

5. Conclusion

This paper proposed a new LooseCut algorithm for interactive image segmentation by taking a loosely bounded box. We further introduced a global similarity constraint and a label consistency term into MRF model. We developed an iterative algorithm to solve the new MRF model. Experiments on three image segmentation datasets showed the effectiveness of LooseCut against several state-of-the-art algorithms. We also showed that LooseCut can be used to enhance the important applications of unsupervised video segmentation and image saliency detection.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010.

[3] S. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *CVPR*, pages 1–8, 2007.

[4] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, pages 3169–3176, 2010.

[5] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, pages 109–122. 2002.

[6] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, pages 105–112, 2001.

[7] M.-M. Cheng, N. Mitra, X. Huang, P. Torr, and S.-M. Hu. Global contrast based salient region detection. *TPAMI*, 37(3):569–582, 2015.

[8] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010.

[9] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, pages 3192–3199, 2013.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.

[11] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009.

[12] Y. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011.

[13] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, pages 277–284, 2009.

[14] H. Li, F. Meng, and K. Ngan. Co-salient object detection from multiple images. *TMM*, 15(8):1896–1909, 2013.

[15] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[16] M. Tang, I. Ayed, and Y. Boykov. Pseudo-bound optimization for binary energies. In *ECCV*, pages 691–707, 2014.

[17] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov. Grabcut in one cut. In *ICCV*, pages 1769–1776, 2013.

[18] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, and Z. Tu. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*, pages 256–263, 2014.

[19] H. Yu, M. Xian, and X. Qi. Unsupervised co-segmentation based on a new global gmm constraint in mrf. In *ICIP*, pages 4412–4416, 2014.

[20] D. Zhang, O. Javed, and M. Shah. Video object co-segmentation by regulated maximum weight cliques. In *ECCV*, pages 551–566. 2014.

[21] Y. Zhou, L.Ju, and S. Wang. Multiscale superpixels and supervoxels based on hierarchical edge-weighted centroidal voronoi tessellation. In *WACV*, pages 1076–1083, 2015.

[22] Y. Zhou, H. Yu, and S. Wang. Feature sampling strategies for action recognition. *CoRR*, abs/1501.06993, 2015.

[23] C. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405. 2014.