MILCut: A Sweeping Line Multiple Instance Learning Paradigm for Interactive Image Segmentation

Jiajun Wu[1*], Yibiao Zhao[2,3*], Jun-Yan Zhu[4], Siwei Luo[2], Zhuowen Tu[5]
[1]ITCS, Institute for Interdisciplinary Information Sciences, Tsinghua University
[2]Beijing Jiaotong University, [3]Department of Statistics, UCLA
[4]Computer Science Division, UC Berkeley, [5]Department of Cognitive Science, UCSD

Abstract

Interactive segmentation, in which a user provides a bounding box to an object of interest for image segmentation, has been applied to a variety of applications in image editing, crowdsourcing, computer vision, and medical imaging. The challenge of this semi-automatic image segmentation task lies in dealing with the uncertainty of the foreground object within a bounding box. Here, we formulate the interactive segmentation problem as a multiple instance learning (MIL) task by generating positive bags from pixels of sweeping lines within a bounding box. We name this approach MILCut*. We provide a justification to our formulation and develop an algorithm with significant performance and efficiency gain over existing state-of-the-art systems. Extensive experiments demonstrate the evident advantage of our approach.*

1. Introduction

Image segmentation is one of the most fundamental problems in computer vision. While fully automated segmentation is arguably an intrinsically ambiguous problem [5] and manual segmentation is time-consuming to obtain, semi-automated (user-interactive) segmentation has demonstrated great practical importance and popularity [10, 32, 15, 9, 22, 29, 19, 8, 35, 27, 20]. Given a moderate level of user input, the goal of interactive segmentation is to segment a foreground object from background based on user input. The system should also be ideally fast enough to have a smooth user interface experience.

Previous interactive segmentation systems including GrabCut [32] and Lazy Snapping [22] have been adopted in multiple domains. However, despite the successes of existing approaches, they all have a noticeable level of limitations. For example, GrabCut [32] and Lazy Snapping [22] are quite efficient but there is still large space for them to improve in accuracy [20, 39]; the pinpointing algorithm [20] combines GrabCut [32] with some heavily engineered initialization priority maps to achieve competitive performances, and their use of approximating algorithms to tackle an NP-hard optimization problem also makes the framework much slower than the Graph Cut algorithm. Apparently both effectiveness and efficiency are essential for a practical interactive image segmentation system.

The type of user input in the current interactive segmentation paradigms can be roughly divided into two categories: scribble based and bounding box based. In general, bounding-box-based interaction gains more popularity because it is more natural for users to provide a bounding box [20]. Also, some methods designed for bounding box interaction can also handle scribbles for refinements [32].

From a different angle, weakly-supervised learning, or specifically multiple instance learning (MIL) [13], has attracted increasing attention in machine learning and many other areas for solving problems with data that contains latent class labels. It tackles the fundamental problem of learning from noisy data by simultaneously learning a classification model and estimating the hidden instance labels. MIL can even be applied to fully-supervised settings due to the intrinsic ambiguity in human annotations. In MIL, instances (data samples) appear in the form of positive and negative bags. The multiple instance constraints request that within each positive bag, there is at least one positive instance; and within the negative bags, all instances are negatives. Therefore, we view MIL as a general formulation for dealing with the hidden class labels in noisy input.

In this paper, we propose a multiple instance learning solution to the interactive image segmentation problem where the property of tight bounding box is explored. An unknown object of interest is supposed to appear within the bounding box at an unknown location; we also know that image pixels outside the bounding box are background. Therefore, we view the image with the bounding box as "noisy input" and our task is to discover the object under weak supervision with the data in company of outliers. Here, we provide a sweeping-line strategy to convert the interactive

$*$ indicates equal contributions.

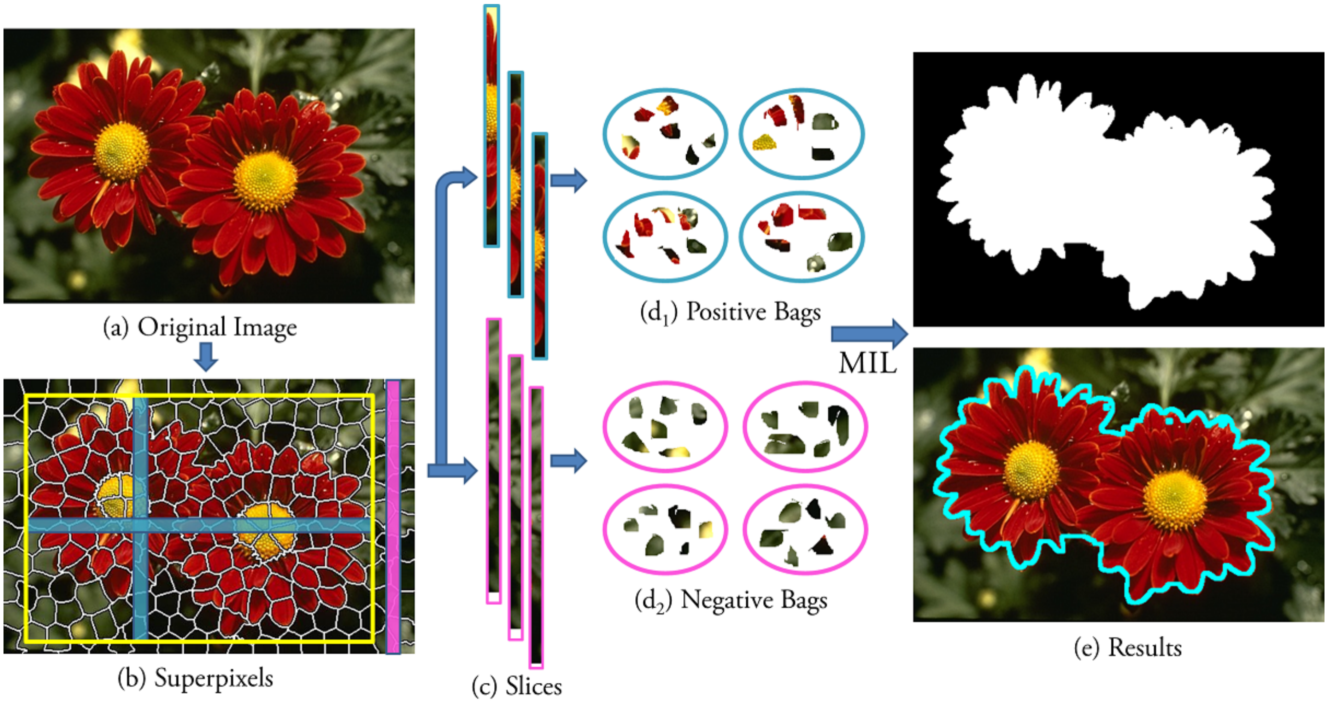

Figure 1: Overview of our MILCut framework for interactive image segmentation. (a) original image, (b) superpixels, (c) slices, (d) positive bags and negative bags, and (e) results.

image segmentation into a multiple instance learning problem. We prove that, under mild conditions, each horizontal or vertical sweeping line of the bounded region must contain at least one pixel from the foreground object, which naturally corresponds to the multiple instance constraints described in the beginning of the section. Therefore, for the task of interactive segmentation with a bounding box, we propose MILCut, a sweeping line multiple instance learning paradigm which uses pixels on the sweeping lines inside the bounding box as positive bags and pixels outside the box as negative bags. Figure 1 provides an overview of our framework. To enforce the topological constraints of foreground objects, we further propose two variants of MILCut in which structural information is explicitly involved.

Our contributions in this paper include: (1) we convert the interactive image segmentation task with bounding box interaction into a multiple instance learning scenario; (2) we propose MILCut, a sweeping line multiple instance learning paradigm, to solve the problem with structural constraints; (3) we exploit local information of superpixels and power of discriminative classifiers, resulting in significant improvement over existing methods in both accuracy and efficiency.

2. Related Work

Interactive image segmentation has attracted numerous interests in the computer vision community [10]. Rother et al. proposed GrabCut [32] which iteratively estimates Gaussian mixture models [9] and then refines results using Graph Cut. Li et al. proposed Lazy Snapping [22] which separates the segmentation into an object marking step and a boundary editing step. Later a large number of interactive segmentation algorithms emerged. Some representative works include geodesic approaches [28, 8], random walk approaches [39, 18], discriminative learning approaches [14, 43], and those methods using various kinds of priors [35, 15].

Different from all these systems, in this paper, we present a multiple instance learning scenario to classify the foreground object inside the box using the bounding box as a weak label. Probably the most relevant work to ours is [20], which proposes to model the task as an integer programming problem using bounding box prior. Our method differs from theirs in two aspects: 1) we propose to convert the task to a weakly supervised learning problem, and solve it discriminatively using sweeping line multiple instance learning, while they focus on approximating the NP-hard integer programming; 2) we make use of superpixels and local informative features in the MIL framework, which improve our results in both accuracy and efficiency.

Multiple instance learning was first introduced by Dietterich et al. [13] for drug activity prediction. Since then, a large number of methods, *e.g.*, DD [23], EM-DD [42], citation-kNN [37], MI-SVM and mi-SVM [4], and IQH [6], have been proposed to solve the multiple instance learning problem. Viola et al. proposed MIL-Boost [36] which applied boosting to multiple instance setting and showed that MIL-Boost could achieve good performance for object detection. In this paper, we propose MILCut, a sweeping line multiple instance learning paradigm which solves interactive segmentation with MIL. When optimizing the likelihood function, MILCut adopts steps in MIL-Boost but extends them to explicitly incorporate structural information.

Structured prediction models like latent structural SVM [40] or hidden CRF [30] have demonstrated their wide applicability in various tasks. There are also several attempts to combine multiple instance learning with structural data. In 2009, Zhou et al. [44] proposed mi-Graph which models instances in the data as non-i.i.d. samples. Later, Deselaers [12] proposed MI-CRF, in which they model bags as nodes and instances as states in CRF. The most recent work is the MILSD formulation from Zhang et al. [41], which adds global smoothness as a graph regularization term to the objective function. Different from all these approaches, in MILCut, we consider the structure of the image from two perspectives: 1) we exploit the tightness of bounding boxes to formulate the multiple instance learning problem using slices as bags; 2) we explicitly enforce structural relations among instances within the formulation.

3. From Bounding Boxes to MIL

A bounding box helps algorithms to focus only on its interior because we assume that the foreground object completely lies in the bounding box. However, most existing frameworks fail to exploit tightness information given by the bounding box, *i.e.*, they did not realize that object boundaries should be close to the user-provided bounding box in all four directions. Recently, Lempitsky et al. [20] incorporated the notion of tightness into their framework, and proposed the pinpointing algorithm for optimization. Their formulation is an NP-hard integer programming and the approximating pinpointing algorithm is not efficient enough for a real-time interaction image segmentation system.

Here we demonstrate that there exists a natural relation between tightness and the multiple instance constraints.

Definition 1. *For an image I, a bounding box B is* valid *if the foreground object O completely lies inside the box,* i.e., *the intersection of the foreground object and the exterior of the bounding box is an empty set, or $(I \backslash B) \cap O = \emptyset$.*

Definition 2. *For an image I, a bounding box B is* tight *if the foreground object O intersects the left, right, top, and bottom border of the bounding box. If we define B_T, B_B, B_L, and B_R as the top, bottom, left, and right border of the bounding box, respectively, then we know that B is tight is equivalent to $O \cap B_T \neq \emptyset, O \cap B_B \neq \emptyset, O \cap B_L \neq \emptyset$, and $O \cap B_R \neq \emptyset$.*

Assuming validity and tightness of the bounding box, we may then convert the image segmentation task into a multiple instance learning problem. Specifically, we treat the horizontal and vertical slices in the bounding box as positive bags and other slices outside the box as negative bags. Either pixels or superpixels could be used as instances. The part (c) and (d) of Figure 1 illustrate the linkage between the bounding box and multiple instance learning.

Lemma 1. *If a bounding box B is valid and tight and the object O inside the bounding box is connected, then the constructed positive and negative bags satisfy multiple instance constraints.*

A proof can be found in supplementary material.

4. Formulation

In this section, we formulate the interactive image segmentation problem in the multiple instance learning framework. As shown in Figure 1, the first step of MILCut is to construct positive and negative bags for multiple instance learning from the user-provided bounding box. We use SLIC superpixels [1] as instances in multiple instance learning. Superpixels [31] have been proved effective in multiple vision tasks [22, 31]. In our setting, using superpixels not only allows us to incorporate a variety of local informative features, but also offers a dramatic speedup.

4.1. Appearance Model

After constructing positive and negative bags, we then apply MIL-Boost [36] to train an appearance likelihood model for distinguishing foreground object from the clutter background. Specifically, the negative log-likelihood of a set of bags is defined as

$$\mathcal{L}_1(h) = -\log \prod_i p_i^{y_i} (1-p_i)^{(1-y_i)} \tag{1}$$

$$= -\sum_i (y_i \log p_i + (1-y_i) \log(1-p_i)), \tag{2}$$

where $y_i \in \{0,1\}$ is the label of bag i and p_i is the probability that bag i is positive in the current model. With a softmax model like generalized mean, p_i can be derived from instance-level probability p_{ij} as $p_i = (\sum_j p_{ij}^r)^{1/r}$, where $p_{ij} = [1+\exp(-y_{ij})]^{-1}$ is the output of a sigmoid function of instance level classification score y_{ij}. Here y_{ij} is computed as a weighted sum of the outputs of the weak classifiers: $y_{ij} = \sum_t \mu_t h^t(x_{ij})$, where x_{ij} is the feature vector of the instance, and $h^t : \mathcal{X} \to \mathcal{Y}$ is a weak classifier, and μ_t is the weight of the t-th classifier.

We then optimize the log-likelihood in the AnyBoost framework [25]. Specifically, when training a new weak classifier, it assigns derivatives of the cost function with respect to changes in the scores as weights on instances, *i.e.*,

$$w_{ij} = \frac{\partial \mathcal{L}_1(h)}{\partial y_{ij}} = \frac{\partial \mathcal{L}_1(h)}{\partial p_i} \frac{\partial p_i}{\partial p_{ij}} \frac{\partial p_{ij}}{\partial y_{ij}}. \tag{3}$$

4.2. Structured Prediction Model

Directly using MIL-Boost fails to incorporate the topological structure of objects. Following CCMIL [38], we explicitly model the structural information in the likelihood function in training and testing in a formulation named MILCut-Struct.

In MILCut-Struct, structural information is explicitly incorporated in the formulation, and the log-likelihood is defined as

$$\mathcal{L}(h) = \mathcal{L}_1(h) + \lambda \mathcal{L}_2(h), \tag{4}$$

where $\mathcal{L}_1(h)$ is the log-likelihood defined in Eqn (1) for the appearance model, and $\mathcal{L}_2(h)$ is the structural constraint. We define $\mathcal{L}_2(h)$ as

$$\mathcal{L}_2(h) = \sum_{i=1}^{n} \sum_{(j,k)\in E_i} v_{ijk} \|p_{ij} - p_{ik}\|^2, \tag{5}$$

where E_i is the set of all neighboring pairs in the i-th bag, and v_{ijk} is the weight of the pair (j,k). Here we define v_{ijk} as the shared boundary length of the j-th and k-th instances in the i-th bag.

Similar to MIL-Boost, the formulation of MILCut-Struct can also be optimized using gradient descent in the AnyBoost framework [25, 38]. In each iteration, the weight w_{ij} of instance j in bag i can be computed as follows (with generalized mean softmax model):

$$w_{ij} = \frac{\partial \mathcal{L}(h)}{\partial y_{ij}} = \frac{\partial \mathcal{L}(h)}{\partial p_i} \frac{\partial p_i}{\partial p_{ij}} \frac{\partial p_{ij}}{\partial y_{ij}}, \tag{6}$$

$$\begin{aligned} \frac{\partial \mathcal{L}(h)}{\partial y_{ij}} &= \frac{\partial \mathcal{L}_1(h)}{\partial y_{ij}} + \lambda \frac{\partial \mathcal{L}_2(h)}{\partial y_{ij}} & (7)\\ &= \frac{\partial \mathcal{L}_1(h)}{\partial p_i} \frac{\partial p_i}{\partial p_{ij}} \frac{\partial p_{ij}}{\partial y_{ij}} + \lambda \frac{\partial \mathcal{L}_2(h)}{\partial p_{ij}} \frac{\partial p_{ij}}{\partial y_{ij}}, & (8) \end{aligned}$$

where

$$\frac{\partial \mathcal{L}_1(h)}{\partial p_i} = \begin{cases} -\dfrac{1}{p_i} & \text{if } y = 1, \\ \dfrac{1}{1-p_i} & \text{if } y = -1, \end{cases} \tag{9}$$

$$\frac{\partial p_i}{\partial p_{ij}} = p_i \frac{(p_{ij})^{r-1}}{\sum_j (p_{ij})^r}, \qquad \frac{\partial p_{ij}}{\partial y_{ij}} = 2p_{ij}(1-p_{ij}), \tag{10}$$

and

$$\frac{\partial \mathcal{L}_2(h)}{\partial p_{ij}} = \sum_{(j,k)\in E_i} 2v_{ijk}(p_{ij} - p_{ik}). \tag{11}$$

The structural constraints enforce the piecewise smoothness in the resulting segments. An alternative way of incorporating structural information without changing the likelihood function (1) is to apply Graph Cut as a post-processing step. Specifically, for each image, we define data terms based on the probability map given by MILCut, and smoothness terms in the same way as GrabCut [32]. We call this variant MILCut-Graph.

5. Experiments

5.1. Setup

Datasets: We conduct experiments on three popular datasets. The first is the well-known GrabCut dataset [32], which contains 50 natural images with ground truth segmentations. The GrabCut dataset has been used as a benchmark in multiple previous works [9, 20, 28, 39]. We use the same bounding boxes as those in [20].

Recently, McGuinness [26] compiled a new benchmark for evaluating interactive segmentation algorithms. The dataset contains 100 distinct objects from the popular Berkeley dataset [24]. These images are selected to represent various segmentation challenges including texture and lighting conditions, and to ensure accuracy, all ground truth segmentations are manually labeled.

The third dataset we used is the Weizmann segmentation database, which is divided into a single object dataset and a double object dataset. Either of them contains 100 gray level images along with ground truth segmentations. Here we use the Weizmann single object dataset.

All the datasets we used are available online. [1] [2] [3]

Superpixels and Features: As mentioned in Section 4, we use SLIC superpixels [1] to generate about 2,400 superpixels per image. For a typical 400×600 image, each superpixel is about 10×10 pixels large. We use average color of all pixels in both RGB color space and L*a*b* color space, and Leung-Malik (LM) Filter Bank [21] as features for superpixels. The number of dimensions of feature vectors for each superpixel is 39 (three for RGB color space, three for L*a*b* color space, and 33 for LM Filter Bank). Generating superpixels and computing all the features take one to three seconds for each image. Note that in practice, we may treat this process as a pre-processing step which can be done before the entire segmentation framework actually starts.

Implementation Details: When generating bags, we shrink the bounding box by 5% to ensure its tightness (see Definition 2), and collect slices inside as positive bags; we then expand the original bounding box by 10% to ensure its validity (see Definition 1), and sample slices outside but close to the expanded bounding box as negative bags. With respect to MILCut-Struct and MILCut-Graph, we use Gaussian weak classifiers and set the maximum number of weak classifiers T to 200. Note that typically the optimization scheme converges with a much smaller number of classi-

[1] http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm

[2] http://kspace.cdvp.dcu.ie/public/interactive-segmentation/

[3] http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/

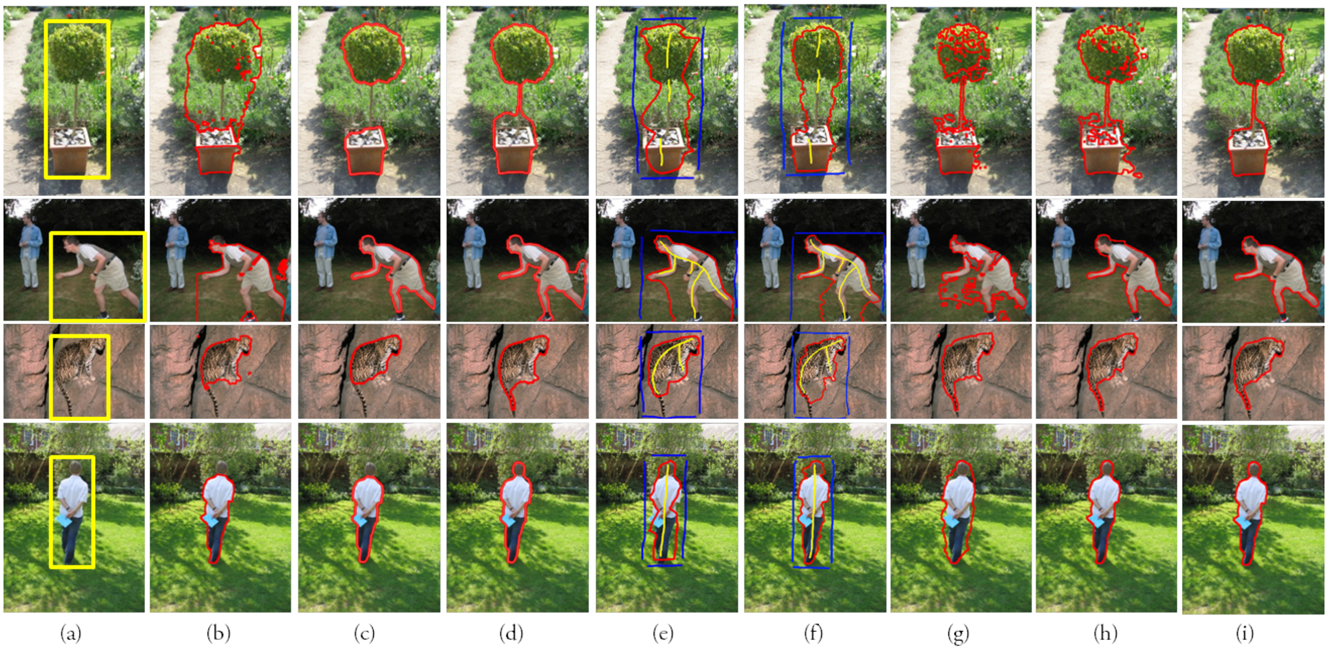

Figure 2: Results of different approaches on the GrabCut dataset. From left to right: (a) original image with the bounding box as input to our formulations and [32, 20], (b) GrabCut [32] (c) GrabCut-GC (InitThirds) [20], (d) GrabCut-Pinpoint (InitThirds) [20], (e) Constrained Random Walker [39], (f) Geodesic Segmentation [8], (g) MILCut without structural information, (h) MILCut-Struct, and (i) MILCut-Graph. As (e) and (f) take scribbles as input, we use yellow scribbles to indicate the user input for foreground and blue ones for background.

Algorithm	Error (%)
MILCut-Struct (ours)	**4.2**
MILCut-Graph (ours)	**3.6**
MILCut (ours)	6.3
Baseline	8.5
LP-Pinpoint [20]	5.0
GrabCut-GC (InitFullBox) [20]	8.9
GrabCut-GC (InitThirds) [20]	5.9
GrabCut-Pinpoint (InitFullBox) [20]	8.2
GrabCut-Pinpoint (InitThirds) [20]	3.7
Simple Interactive Object Extraction [16]	9.1
GMMRF [9]	7.9
Graph Cut [10]	6.7
Lazy Snapping [22]	6.7
Geodesic Segmentation [8]	6.8
Random Walker [18]	5.4
Transduction [14]	5.4
Geodesic Graph Cut [28]	4.8
Constrained Random Walker [39]	4.1

Table 1: Error rates of different approaches on the GrabCut dataset. The first nine algorithms take a single bounding box as input, while the others use a “Lasso” form of trimaps which contain richer information. InitFullBox and InitThirds are two ways of initialization used in [20].

Algorithm	Jaccard Index (%)
MILCut-Struct (ours)	**84**
MILCut-Graph (ours)	**83**
MILCut (ours)	78
Baseline	76
GrabCut [32]	77
Binary Partition Trees [33]	71
Interactive Graph Cut [10]	64
Seeded Region Growing [2]	59
Simple Interactive Object Extraction [16]	63

Table 2: Jaccard Indices on the Berkeley dataset. The last four methods take scribbles as input.

Algorithm	ours	[7]	[3]	[17]	[34]	[11]
F-score (%)	**0.89**	0.87	0.86	0.83	0.72	0.57

Table 3: F-scores on the Weizmann single object dataset.

fiers. Generalized mean is used as the softmax model, and the exponent r is set to 1.5. In MILCut-Struct, we set λ to 0.05. Following [32], we finally apply border matting on the probability map produced by MILCut in order to refine the results by deriving smooth object boundaries.

Metrics: Following [28, 20, 39], we use error rates as the

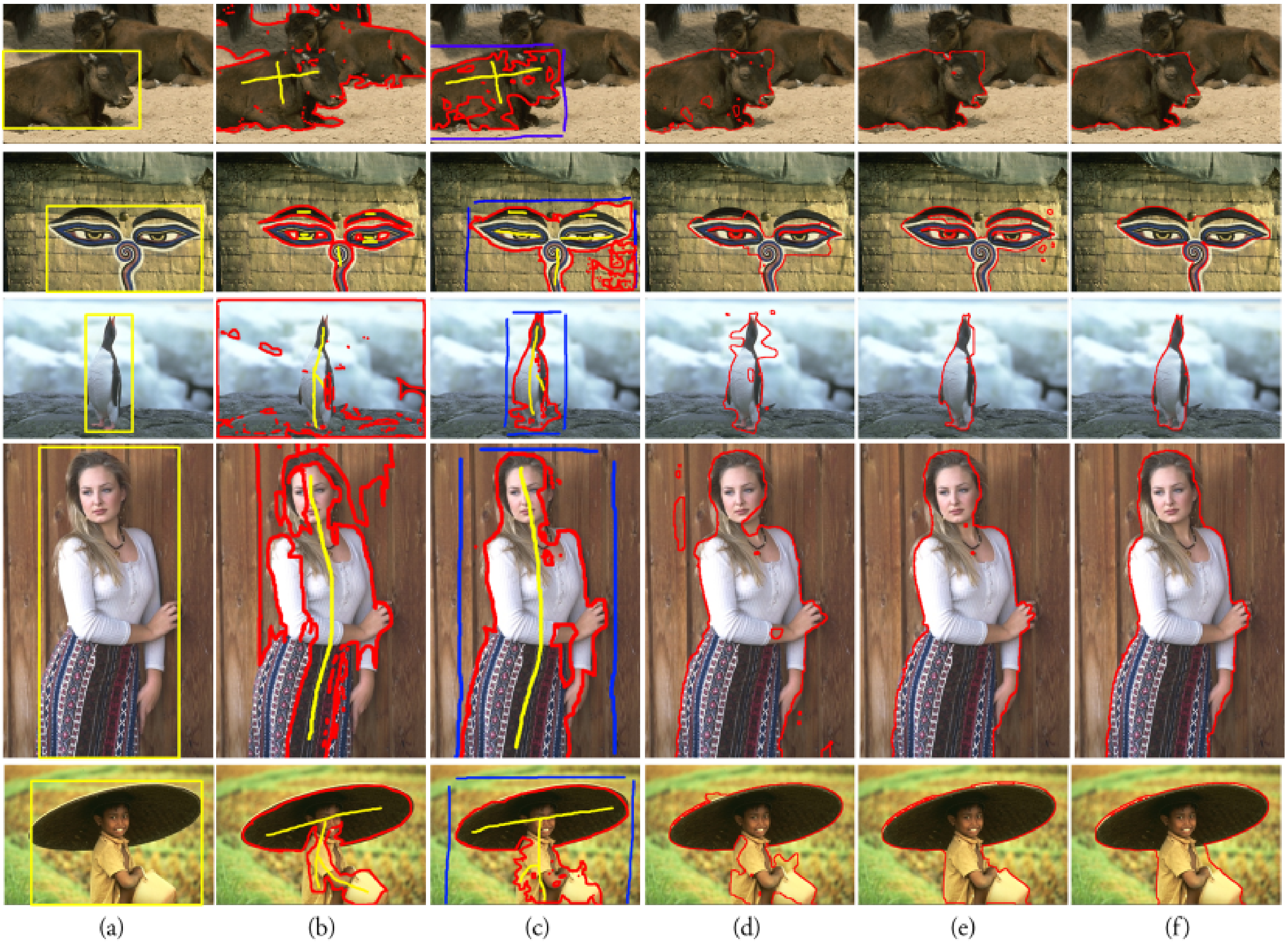

Figure 3: Results of different approaches on the Berkeley dataset. From left to right: (a) original image with the bounding box as input to our formulations, (b) Seeded Region Growing [2], (c) Simple Interactive Object Extraction [16], (d) MILCut without structural information, (e) MILCut-Struct, and (f) MILCut-Graph. As (b) and (c) take scribbles as input, we use yellow scribbles to indicate the user input for foreground and blue ones for background.

metric for measuring accuracy of the output on the GrabCut dataset. The error rate is defined as the ratio of number of misclassified pixels to number of pixels in unclassified region. For Berkeley dataset, we follow [26], which uses binary Jaccard index to measure the object accuracy. The measure is given by $J = |G \cap M|/|G \cup M|$, where G is the ground truth and M is the output.

5.2. Results and Discussions

GrabCut: The results for the GrabCut dataset are shown in Table 1, and Figure 2 illustrates some of the outputs of different algorithms. Some results are reported by previous works [20, 28, 39, 9]. For comparison, we also conduct experiments using MILCut without considering structural information, and a baseline approach which uses all superpixels inside the bounding box as a single positive bag in MIL without applying the sweeping line paradigm. We observe that although purely using MIL does not guarantee promising performances (8.5% Baseline), exploiting the relationship between multiple instance learning and the tightness of the bounding boxes provides competitive results (6.3%, MILCut), and adding structural constraints further reduces the error rate to 4.2% (MILCut-Struct) or 3.6% (MILCut-Graph). We can see that the sweeping line paradigm and the explicitly enforced structural constraints help our algorithm outperform the other state-of-the art methods. The only algorithms which are barely comparable with ours include GrabCut-Pinpoint (InitThirds) [20] and Constrained Random Walker [39]. However, they either employ heavily engineered initializations which make their approach at least ten times slower than ours, or use trimaps as input, in which most of the pixels are already labeled and only a strip of pixels along boundaries are left for inference.

Berkeley: For the Berkeley dataset, we can see from

Table 2 and Figure 3 that, again, all of our approaches outperform the other reported algorithms. The results of the other methods first appeared in [26]. Note that our approach uses bounding boxes as input while four others use more time-consuming scribbles. Specifically, Seeded Region Growing [2] requires time-consuming parameter tuning and generates unsatisfactory segments due to its lack of use of background scribbles. Simple Interactive Object Extraction [16], without exploiting the structural information of objects, fails to produce coherent foreground objects.

To fairly compare all algorithms, we invited four external graduate students to provide scribbles and for each image, we use input (bounding boxes or scribbles) provided by them within 10 seconds. In [26], it is reported that if an user could spend on each image about 60 seconds to provide additional scribbles, then the performances of the other algorithms can be boosted to about 90%. For our framework, it is also natural to incorporate additional user input: we could simply assign superpixels related to positive/negative scribbles as positive/negative instances. We experimented and found that if we could spend about 10 more seconds to provide about three scribbles to each image, the Jaccard Index of our framework quickly increases to over 90%. In this case, we set the maximum number of weak classifiers to $T = 15$ to provide real-time feedback.

Weizmann: On the Weizmann single object dataset, as shown in Table 3, MILCut-Graph also consistently ourperforms all other widely-used segmentation algorithms [7, 3, 17, 34]. The results of others are taken from the website of the Weizmann segmentation database.

In general, MILCut explicitly embeds the bounding box prior in the model, and is able to stretch the foreground segment towards all sides of the bounding box. This is illustrated by the heads in the second and fourth rows of images in Figure 2. The two variants of our approach, MILCut-Struct and MILCut-Graph, are generally comparable as shown in both Figure 2 and 3. On one hand, the Graph Cut algorithm in MILCut-Graph performs global optimization while MILCut-Struct uses gradient descent in the feature space which might fall in local minima. On the other side, MILCut-Struct naturally embraces the MIL learning paradigm, strengthening the appearance model in training.

Running Time: For each image, after superpixels are generated and features are computed in one to three seconds altogether, both MILCut-Struct and MILCut-Graph take only one to two seconds to segment a foreground object. Also, it takes only five to ten seconds for either MILCut-Struct or MILCut-Graph to segment an image based on pixels instead of superpixels. Comparatively, the systems whose accuracies are competitive to ours, GrabCut-Pinpoint [20] and Constrained Random Walker [39], both have certain limitations. GrabCut-Pinpoint [20] needs to model Gaussian mixture models as an initialization, and then iteratively solve an integer programming using pinpointing for five times. The whole system takes several minutes to segment an image, and even with GPU acceleration, it is one or two orders of magnitude slower than our method. Constrained Random Walker [39] has comparative accuracy and running time with our approach, but takes trimaps as input, which are harder to obtain and contain much richer information than bounding boxes.

	MILCut-Graph	GrabCut [32]
No noise	0.89	0.88
Human noise	0.89	0.86
Machine noise	0.86	0.85

Table 4: F-scores on the Weizmann single object dataset with noisy input.

Algorithm	ours	[3]	[17]	[11]	[34]
F-score (%)	**0.71**	0.68	0.66	0.61	0.58

Table 5: F-scores on the Weizmann double object dataset.

5.3. Experiments with Noisy Input

In real cases, the assumptions we made for MILCut cannot always be satisfied. In this section, we consider two distinct situations where multiple instance constraints are not met: 1) The bounding box is not tight; 2) The object is not connected. Experiments show that MILCut can still obtain better performance than other approaches in these cases.

Inaccurate Bounding Boxes: We consider two types of inaccurate bounding boxes. In one case, we invite graduate students to label each image within five seconds, leading to noisy and inaccurate boxes (named *human noise*). In a second experiment, for each image, we scale the distances between the center of the box and each of its four sides with coefficients randomly sampled in $[0.8, 1.4]$ (named *machine noise*). In both cases, the MIL constraints are no longer strictly satisfied. We can see from Table 4 that MILCut-Graph can do better than GrabCut [32] consistently.

Unconnected Objects: We then apply our algorithm on the Weizmann double object dataset, where in each image, there are two unconnected objects. We use a single bounding box containing both objects as input to our approach. The results are shown in Table 5. Our approach MILCut-Graph achieves a higher F-score than all other listed methods. Here the results of others are taken from the website of the Weizmann segmentation database.

6. Conclusion

In this paper, we propose MILCut, a novel sweeping line multiple instance learning paradigm to segment the foreground object inside a user-provided bounding box. We theoretically justify that the sweeping line strategy for MIL bagging naturally embraces the user's intention carried by a bounding box. The algorithm is simple, easy to implement, and shown to be a powerful tool with superior accuracy and efficiency. We believe our observation of modeling interactive image segmentation in a multiple instance learning setting would encourage future research in both areas.

Acknowledgments

This work was supported by NSF IIS-1216528 (IIS-1360566) and NSF CAREER award IIS-0844566 (IIS-1360568). Jiajun Wu is supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2282, 2012. 3, 4

[2] R. Adams and L. Bischof. Seeded region growing. *TPAMI*, 16(6):641–647, 1994. 5, 6, 7

[3] S. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *CVPR*, 2007. 5, 7

[4] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002. 2

[5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011. 1

[6] B. Babenko, N. Verma, P. Dollár, and S. J. Belongie. Multiple instance learning with manifold bags. In *ICML*, 2011. 2

[7] S. Bagon, O. Boiman, and M. Irani. What is a good image segment? a unified approach to segment extraction. In *ECCV*, 2008. 5, 7

[8] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV*, 82(2):113–132, 2009. 1, 2, 5

[9] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, pages 428–441, 2004. 1, 2, 4, 5, 6

[10] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, 2001. 1, 2, 5

[11] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24(5):603–619, 2002. 5, 7

[12] T. Deselaers and V. Ferrari. A conditional random field for multiple-instance learning. In *ICML*, 2010. 3

[13] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997. 1, 2

[14] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, and F. Ségonne. Segmentation by transduction. In *CVPR*, 2008. 2, 5

[15] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, 2005. 1, 2

[16] G. Friedland, K. Jantz, and R. Rojas. Siox: Simple interactive object extraction in still images. In *ACM Multimedia*, 2005. 5, 6, 7

[17] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *ICCV*, pages 716–723, 2003. 5, 7

[18] L. Grady. Random walks for image segmentation. *TPAMI*, 28(11):1768–1783, 2006. 2, 5

[19] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, 2010. 1

[20] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *CVPR*, 2009. 1, 2, 3, 4, 5, 6, 7

[21] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001. 4

[22] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Transactions on Graphics*, 23(3):303–308, 2004. 1, 2, 3, 5

[23] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *NIPS*, 1998. 2

[24] 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. In *ICCV*, 2001. 4

[25] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space. In *NIPS*, 1999. 3, 4

[26] K. McGuinness and N. E. O'Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010. 4, 6, 7

[27] E. N. Mortensen and W. A. Barrett. Interactive segmentation with intelligent scissors. *Graphical models and image processing*, 60(5):349–384, 1998. 1

[28] B. L. Price, B. Morse, and S. Cohen. Geodesic graph cut for interactive image segmentation. In *CVPR*, 2010. 2, 4, 5, 6

[29] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *TIP*, 16(4):1046–1057, 2007. 1

[30] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *TPAMI*, 29(10):1848–1852, 2007. 3

[31] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003. 3

[32] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. 1, 2, 4, 5, 7

[33] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *TIP*, 9(4):561–576, 2000. 5

[34] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000. 5, 7

[35] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008. 1, 2

[36] P. Viola, J. Platt, C. Zhang, et al. Multiple instance boosting for object detection. In *NIPS*, 2006. 2, 3

[37] J. Wang and J.-D. Zucker. Solving the multiple-instance problem: A lazy learning approach. In *ICML*, 2000. 2

[38] Y. Xu, J.-Y. Zhu, E. I. Chang, M. Lai, and Z. Tu. Weakly supervised histopathology cancer image segmentation and classification. *MIA*, 18(3):591–604, 2014. 4

[39] W. Yang, J. Cai, J. Zheng, and J. Luo. User-friendly interactive image segmentation through unified combinatorial user inputs. *TIP*, 19(9):2470–2479, 2010. 1, 2, 4, 5, 6, 7

[40] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009. 3

[41] D. Zhang, Y. Liu, L. Si, J. Zhang, and R. D. Lawrence. Multiple instance learning on structured data. In *NIPS*, 2011. 3

[42] Q. Zhang and S. A. Goldman. Em-dd: An improved multiple-instance learning technique. In *NIPS*, 2001. 2

[43] Y. Zhao, S.-C. Zhu, and S. Luo. Co3 for ultra-fast and accurate interactive segmentation. In *ACM Multimedia*, 2010. 2

[44] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li. Multi-instance learning by treating instances as non-iid samples. In *ICML*, 2009. 3