FINE-STRUCTURED OBJECT SEGMENTATION VIA EDGE-GUIDED GRAPH CUT WITH INTERACTION SIMPLIFICATION

Yongchao Gong, Shiming Xiang, Lingfeng Wang and Chunhong Pan

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
{yongchao.gong, smxiang, lfwang and chpan}@nlpr.ia.ac.cn

ABSTRACT

Fine-structured object segmentation is a challenging problem in object segmentation community. There are mainly two difficulties that can seriously degrade the segmentation quality: 1) insufficient interactions on fine structures due to the high demand of time and manual efforts, and 2) shrinking bias that discourages long object boundaries. To address these two issues, we develop a novel method within the graph cut framework. First, the commonly used operation of scribbling or dragging bounding boxes is replaced by loosely drawing a few rectangles, thus the interaction burden is largely reduced. Second, an edge-guided graph cut model is proposed to mitigate shrinking bias. This model enforces connectivity of fine structures by adjusting the weighting between neighboring pixels. Finally, the segmentation task is formulated as an optimization problem, which can be optimized effectively and efficiently. Comparative experimental results demonstrate the effectiveness of our method.

***Index Terms*—** Fine-structured object segmentation, interaction simplification, edge-guided graph cut

1. INTRODUCTION

In recent years, fine-structured (FS) object segmentation has attracted increasing attention. In practical applications, *e.g.*, image synthesis and plant modeling, there is an urgent demand of effective approaches for this task. Despite the progress in general object segmentation, many notable approaches [1–8] fail to handle FS objects well, thus this challenging task remains far from being well solved.

In object segmentation community, graph cut (GC) [2, 9] has proved to be a popular and powerful technique. It formulates segmentation problem on a Markov random field (MRF) [10] and makes exact inference efficiently via mincut/maxflow algorithm [11]. However, two major difficulties can seriously degrade the semantic completeness and connectivity of the segmented objects [12–16]. 1) **Insufficient interactions** make the appearance models learned from the interactions unreliable. This is because scribbling on fine structures with small width requires a large amount of time and manual efforts for annotators [2, 12–14]. Bounding boxes [4] are proposed later to replace the operation of scribbling and reduce the interaction burden, but they have to enclose the whole object, and turn out to be meaningless when the object (*e.g.*, a tree) is spanning over the entire image. 2) **Shrinking bias** [12–17] tends to discourage fine parts with long boundaries. Intuitively, the latter difficulty can be overcome by providing sufficient interactions (see Fig. 1), but the former difficulty makes it a tough task. To mitigate shrinking bias, various methods have been proposed, which can be roughly categorized into topological constraints based and cooperative cuts based.

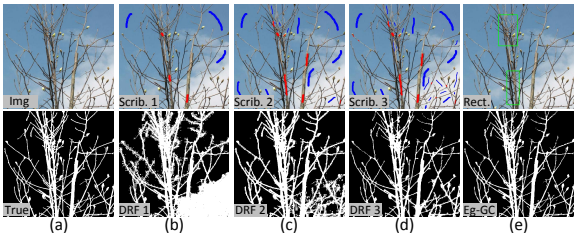

Fig. 1. Relationship between shrinking bias and the required amount of interactions in fine parts. (a) Input image and ground truth. (b-d) Interacting with increasing amount of scribbles (red-object, blue-background) and results obtained by the method in [14]. (e) Interacting with rectangles (green) and result obtained by our method. As demonstrated, a satisfactory result requires time-consuming scribbles, while our method is able to overcome this drawback.

Most topological constraints based methods try to introduce appropriate priors, *e.g.*connectivity [12], or tree shape [18], into their models. These priors enforce the connectivity of fine parts to the main body. However, most of these methods suffer from NP-hardness. As only approximate algorithms are developed, global optima can not be guaranteed. Moreover, since the methods in [12] and [18] require extra scribbles for each fine part, they are impractical to handle complex objects with many fine parts.

Recently, cooperative cuts (CC) [13] has shown the power to mitigate shrinking bias. The core idea of CC is to reduce the cut cost of long boundaries by reweighting the graph edges[1] according to the similarities between them [13]. To minimize the resulting non-submodular [19] energy function, an iterative strategy is adopted in [13] but only an approximate solution can be achieved. Later, Kohli *et al.* [14] reformulated this model as a higher-order MRF [20], or a deep random field (DRF) and developed a globally optimal algorithm, but the inference is quite time-consuming.

To address the two aforementioned difficulties, we propose an edge-guided GC (Eg-GC) model with interaction simplification. Specifically, we provide interactions with simply a few loosely drawn rectangles[2] rather than the commonly used operation of scribbling or dragging bounding boxes. Moreover, to better capture the actual object boundaries, we adaptively adjust the weighting factor for each graph edge individually instead of equally treating them over the entire image. Experimental results demonstrate the

This work was supported by the National Natural Science Foundation of China under Grants 61370039, 61272331 and 61331018, and the Beijing Natural Science Foundation under Grant 4142057.

[1]For clarity, 'graph edge' denotes an edge linking two neighbouring pixels, while 'edge' alone refers to image edge, *e.g.*, Canny edge.

[2]Note that a 'rectangle' does not have to enclose the entire object, which is different from 'bounding box' proposed in [4].

effectiveness of our method for segmenting FS objects.

The main advantages of our method are listed as follows:

1. A user-friendly and time-saving interaction paradigm is proposed, which largely reduces the required amount of time and manual efforts. Meanwhile, comparable or even better performance is achieved in comparison to methods using scribbles as interactions.

2. The Eg-GC model enforces a more reasonable weighting between the unary and pairwise terms in the energy function of GC. This contributes to enhance the completeness and connectivity of the FS objects, thus mitigating shrinking bias.

3. With both plentiful seeds generated from the simplified interactions and the Eg-GC model, the segmentation task is formulated as an energy minimization problem, in which the global optimum can be achieved effectively and efficiently.

2. THE PROPOSED METHOD

2.1. A brief review on graph cut

Given an image $\mathcal{I}$, let $\mathcal{V}$ be the set of pixels and $\mathcal{E}$ be the set of graph edges. The RGB feature vector of pixel i is denoted by $\mathbf{p}_i \in \mathbb{R}^3$. A labeling vector $\mathbf{x} = \{x_1, x_2, ..., x_n\} \in \mathbb{R}^n$ (n is the number of pixels in $\mathcal{I}$) denotes the set of binary indicator variables of the pixels, taking label '1' for object and '0' for background. The goal of graph cut (GC) [2, 9] is to solve $\mathbf{x}$ by minimizing the following energy:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} U_i(x_i) + \lambda \sum_{(i,j) \in \mathcal{E}} V_{ij}(x_i, x_j) \ , \tag{1}$$

where $U_i(\cdot)$ is the unary potential of pixel i and $V_{ij}(\cdot, \cdot)$ is the pairwise potential of pixels i and j. The positive weighting factor λ is used to specify the relative importance between the two terms. It is tuned *globally* to yield satisfactory results.

The function $U_i(x_i)$ indicates the labeling preference for pixel i. The function $V_{ij}(x_i, x_j)$ encodes the similarity of two neighbouring pixels i and j, which is defined as $V_{ij}(x_i, x_j) = w_{ij}||x_i - x_j||_1$. Here w_{ij} is the penalty for label discontinuity between pixels i and j, and $||\cdot||_1$ denotes ℓ_1-norm.

The global minimum of $E(\mathbf{x})$ can be achieved by calling the mincut/maxflow algorithm [11]. However, although GC is effective in handling compact objects, its effectiveness for FS objects is prone to degrade due to the aforementioned difficulties, *i.e.*, insufficient interaction and shrinking bias. Therefore, we propose a novel Eg-GC model with interaction simplification to overcome these difficulties.

2.2. Interaction simplification

Since FS objects are hard to provide interactions by either scribbling or drawing bounding boxes (as analyzed in Section 1), we propose a user-friendly and time-saving paradigm for simplifying user interactions. In this paradigm, a user only needs to loosely drag one or a few rectangles to indicate the regions where foreground pixels distribute relatively densely. Then a novel iterative refining scheme is developed to generate foreground and background seeds from the interactions. These seeds will work as an alternative of the commonly used user-specified scribbles in the subsequent processing.

Next we detail the iterative scheme. Each iteration consists of foreground probability estimation, seeds generation and foreground probability refinement. Since pixels with similar RGB features tend to have similar foreground probabilities, before the iterations start, K-means clustering is first performed to cluster pixels in $\mathcal{I}$ into K subsets, denoted by $\{\mathcal{C}_k\}_{k=1}^K$. In this manner, pixels in the same cluster can be assigned with the same foreground probability estimate. In addition, we denote $\mathcal{R}^{(1)}$ (the superscript indicates the iteration time starting from $t = 1$, the same below) as the set of pixels in the user-specified rectangles, and it will be updated in the iterations.

Foreground probability estimation. For the k-th subset $\mathcal{C}_k$, we define *local density* and *global density* as the densities of foreground pixels in the rectangles and in the entire image, respectively, namely,

$$D_{\text{local}}(k) = |\{p | p \in \mathcal{C}_k \cap \mathcal{R}^{(t)}\}| / |\mathcal{R}^{(t)}| \ , \tag{2}$$

$$D_{\text{global}}(k) = |\{q | q \in \mathcal{C}_k \cap \mathcal{V}\}| / |\mathcal{V}| \ , \tag{3}$$

where pixels p and q are from $\mathcal{R}^{(t)}$ and $\mathcal{V}$, respectively. Throughout this paper, $|\cdot|$ denotes the number of elements in a set. For an arbitrary pixel i in $\mathcal{C}_k$, we define the ratio of local and global densities as

$$R_{\text{LG}}(i) = D_{\text{local}}(k) / D_{\text{global}}(k), \forall i \in \mathcal{C}_k, k = 1, 2, \ldots, K, \tag{4}$$

According to the specification of $\mathcal{R}^{(t)}$, for a foreground pixel $i \in \mathcal{C}_k$, $D_{\text{local}}(k)$ is much higher than $D_{\text{global}}(k)$, *i.e.*, $R_{\text{LG}}(i)$ has a large value. Therefore, the larger $R_{\text{LG}}(i)$ is, the more likely pixel i tends to be a foreground pixel. After calculating all the $R_{\text{LG}}(i), \forall i \in \mathcal{I}$, we linearly scale them to be in the range $[0, 1]$ as the estimate of foreground probability of each pixel. In this way, we obtain the estimated foreground probability map $\mathbf{P}^{(t)}$, with the same size as $\mathcal{I}$.

Seed generation. To generate seeds from $\mathbf{P}^{(t)}$, a Gaussian filter is first applied to spatially smooth $\mathbf{P}^{(t)}$, denoted by $\mathbf{P}_S^{(t)}$. This can also reduce the probable influence of noise. In $\mathbf{P}_S^{(t)}$, if the value at pixel i is still equal to 1, we can ensure that all the pixels in the local patch centered at pixel i have the same foreground probability with pixel i in $\mathbf{P}^{(t)}$. For this reason, pixel i is likely to be a foreground pixel, thus can be extracted as a foreground seed. In this way, we extract the set of foreground seeds $\mathcal{F}^{(t)}$. Similarly, we can also extract the set of background seeds $\mathcal{B}^{(t)}$ if their values in $\mathbf{P}_S^{(t)}$ are equal to 0.

Foreground probability refinement. Once we obtain $\mathcal{F}^{(t)}$ and $\mathcal{B}^{(t)}$, we use them as positive and negative samples, respectively, to train a logistic regression model. Afterwards, for each pixel i, a probabilistic value is obtained via this model, namely,

$$\Pr(x_i = 1 | \mathbf{p}_i) = 1 / (1 + \exp(-(\mathbf{w}^{\mathrm{T}} \mathbf{p}_i + b))) \ , \tag{5}$$

to measure its probability of being classified as foreground. Parameters $\mathbf{w}$ and b are learned in the training procedure. In this way, we obtain a refined foreground probability map. Since we want to generate more reliable seeds in the next iteration, a more accurate $\mathcal{R}^{(t+1)}$ generated by adopting seed generation on this refined map is used to replace $\mathcal{R}^{(t)}$ in Eq. (2). This is because $\mathcal{R}^{(t+1)}$ is generated from a refined map, thus $\mathcal{R}^{(t+1)}$ is closer to the actual FS object than $\mathcal{R}^{(t)}$.

We then iteratively perform the above three steps, until the absolute value of $(|\mathcal{R}^{(t+1)}| - |\mathcal{R}^{(t)}|)/|\mathcal{R}^{(t)}|$ is below a pre-specified threshold τ. Note that this process is inspired by sample selection using active learning in machine learning [21]. Each iteration helps to make the seeds more reliable, thus in turn improves the accuracy of the probability. In our experiments, convergence is always achieved within at most three iterations when $\tau = 10^{-4}$. Upon convergence, we adopt seed generation again on the finally refined map to extract $\mathcal{F}$ and $\mathcal{B}$ as the final foreground and background seeds.

If parts of the background are very similar to the foreground, they will be wrongly included in $\mathcal{F}$ and can not be excluded automatically. To impose user's understanding about the object, we adopt a further user editing mechanism to enable the user to manually exclude these parts with a few additional rectangles. This mechanism is inspired by GrabCut [4] to tackle difficult images, while other images do not have the necessity. Even so, it should be noted that the total amount of interactions is still far less than scribbling.

2.3. The edge-guided graph cut model

An inherent drawback in Eq. (1) lies in that the two terms are simply weighted by a single factor λ, as most GC based methods do [2, 4,

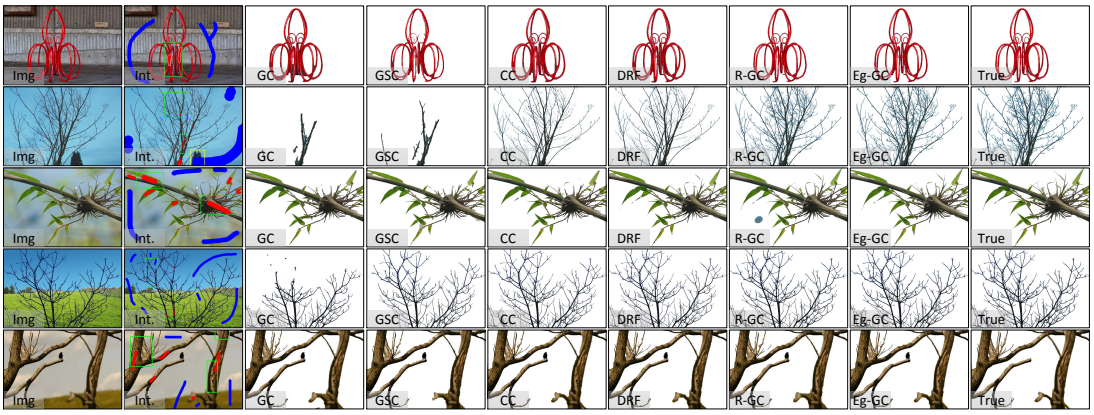

Fig. 2. Qualitative results of images in the 'FineStruct' dataset. Source images are shown in the first column. Scribbles (red-object, blue-background) are used for GC, GSC, CC and DRF, and rectangles (green) for R-GC and Eg-GC, as shown in the second column. They are put together for illustration due to the space limitation. To avoid color confusion, the foreground scribbles for the red object in the first row are highlighted with white boundaries. Note that in the second row further user editing is adopted due to the ambiguity, as indicated by the yellow rectangle. From the third to the last column are the results obtained by GC, GSC, CC, DRF, R-GC and Eg-GC, and the ground truth.

12–16]. It is equivalent to enforcing equal importance to cut all the graph edges, thus fine parts with long boundaries are discouraged due to the large cut cost. This drawback is known as shrinking bias [12–17]. To overcome this drawback, an edge-guided graph cut (Eg-GC) model is developed to selectively encourage the cut of each graph edge with an adaptive weighting factor.

Now we present the details of the Eg-GC model. According to the RGB difference of the two endpoints, all the directed graph edges are first grouped into the N bins of color histogram $\mathcal{H}$. This enables us to equally treat graph edges in the same bin due to the appearance similarity. In the h-th bin, the graph edges share a common weighting factor μ_h which is specified according to image edge information (as will be detailed later). In this way, the second term in Eq. (1) can be reformulated as the group-wise weighted sum of the pairwise potentials, thus yielding the energy function of Eg-GC:

$$E_{\text{new}}(\mathbf{x}) = \sum_{i\in\mathcal{V}} U_i(x_i) + \lambda \sum_{h=1}^{N} (\mu_h \sum_{(i,j)\in\mathcal{E}_h} V_{ij}(x_i, x_j)), \quad (6)$$

where $\mathcal{E}_h$ ($h = 1, 2, \ldots, N$) denotes the h-th bin of $\mathcal{H}$. To impose a selective weighting for each bin, an individual μ_h is assigned to the graph edges in $\mathcal{E}_h$. It is worth noting that in a specific problem, the second term is still globally scaled by a positive constant λ. In addition, although the first term has the same definition as that in Eq. (1), it becomes more accurate now because more seeds (in comparison to those provided by scribbles) are used to model the likelihood.

To specify μ_h, the edge map Ψ generated by Canny detector [22] is used as effective guidance for the calculations. We group the graph edges in $\mathcal{E}_h$ with one endpoint located on Ψ as a set $\mathcal{E}_h^{\Psi}$ and define ratio $p_h = |\mathcal{E}_h^{\Psi}|/|\mathcal{E}_h|$. Intuitively, cutting the graph edges with endpoints located on Ψ should be encouraged. If some of the edges have been wrongly detected, *e.g.*, due to texture regions or noise, p_h tends to be very low. For this reason, if p_h is high enough (we call $\mathcal{E}_h$ an *active* bin), *i.e.*, higher than a specified threshold γ, the corresponding edges can be viewed as a reasonable estimate of the actual object boundaries. In such a bin $\mathcal{E}_h$, to make the graph edges more likely to be cut, we lower their weightings in order to reduce the cost of cutting them. That is to say, the more graph edges we want to cut, the more the weightings should be lowered, thus preventing the cost from becoming too large. Based on the above analyses, we set the weighting factor μ_h for graph edges in $\mathcal{E}_h$ as

$$\mu_h = \begin{cases} 1/|\mathcal{E}_h| \,, & \text{if } p_h > \gamma \\ 1 \,, & \text{if } p_h \le \gamma \,. \end{cases} \quad (7)$$

Although the second term in Eq. (6) has been modified in comparison to that in Eq. (1), its submodularity [19] remains unchanged, thus the global optimum of $E_{\text{new}}(\mathbf{x})$ is still guaranteed. To ensure the effectiveness and efficiency, the minimization is performed via calling the mincut/maxflow algorithm [11].

Due to the weighting adjustment, the total cost of cutting the graph edges in an *active* bin $\mathcal{E}_h$ decreases from $\sum_{(i,j)\in\mathcal{E}_h} V_{ij}(x_i, x_j)$ to $1/|\mathcal{E}_h| \sum_{(i,j)\in\mathcal{E}_h} V_{ij}(x_i, x_j)$, which is equivalent to imposing a discount. By this means, the cut of fine parts with long boundaries no longer costs as much as before, thus shrinking bias is effectively mitigated, making Eg-GC capable to handle FS objects.

Note that Canny edges are not always accurate enough. However, in this model they are only used as implicit constraints, and can be interpreted as *a priori* knowledge of boundary location. For this reason, only a coarse estimate of the actual object boundaries is enough to yield satisfactory results, and accuracy is not strictly required here. In comparison to using explicit topological constraints [12, 18], this strategy helps to reduce the model complexity and avoid NP-hardness.

3. EXPERIMENTS

In this section, we apply the proposed method to FS object segmentation. Comparisons are made with several representative methods.

Dataset description. To highlight the effectiveness of the proposed interaction paradigm, we collected a challenging dataset for evaluation called 'FineStruct', where all the images are difficult to provide scribbles (requiring many zooming in and out operations) due to the fine structures and a lack of compact main bodies. It consists of 30 images: 22 images with manually labeled groundtruth, and 8 representative images from the 'Twigs&Legs' dataset [13,23]. In addition, we made evaluations on the 'MSRC' dataset [24] only for qualitative comparisons due to the absence of detailed ground truth. Furthermore, images containing compact objects from this dataset are also considered in order to ensure that the performance of our method does not degrade in segmenting compact objects.

Methods for comparisons. We compare the performance of *six* methods. GC [2, 25] is the baseline. Three representative GC based methods in FS object segmentation, *i.e.*, geodesic star convexity

Table 1. Quantitative comparisons for the results in Fig. 2 (Img1–Img5 correspond to the five images from the 'FineStruct' dataset) and *averaged over the whole dataset* (Avg.). Parameters are chosen in such a way that the averaged IOU over the whole dataset is the highest, namely $K = 5, N = 32^3, \gamma = 0.08$ and $\lambda = 0.1$. The values in red indicate the best and blue the second best. Best viewed in color.

Crit.	IoU (%)						Err. (%)					
Meth.	GC	GSC	CC	DRF	R-GC	Eg-GC	GC	GSC	CC	DRF	R-GC	Eg-GC
Img1	84.7439	85.6911	88.4426	88.6191	92.2203	92.4324	2.1517	1.7425	1.5409	1.5131	0.3126	0.5214
Img2	14.4057	16.3375	52.1862	52.1772	66.5211	67.5397	6.7293	6.2116	0.3739	0.3789	0.0518	0.0651
Img3	88.3719	85.8220	88.7573	89.1896	90.7222	92.3063	2.7821	3.4094	2.6773	2.5588	2.2660	1.8227
Img4	62.7808	77.3188	82.0690	82.0355	89.6901	93.4571	4.8018	2.9313	2.3163	2.3205	1.3327	0.8470
Img5	92.5609	90.3749	92.7864	92.8522	92.8552	92.9316	2.3700	3.0413	2.2743	2.2501	2.2507	2.2426
Avg.	72.6972	74.0203	75.3864	78.4925	79.9210	81.1656	4.4348	4.4054	4.0282	3.4484	2.9488	2.6244

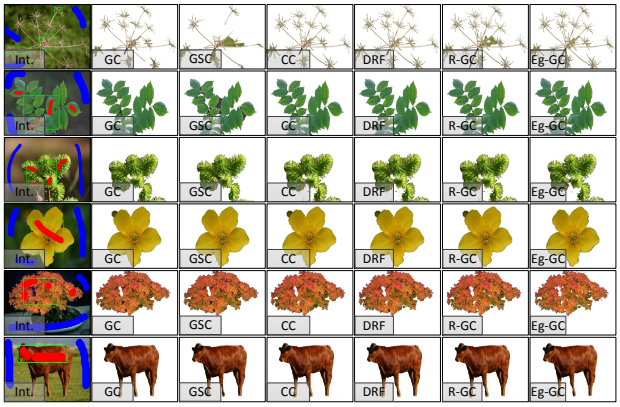

Fig. 3. Qualitative results of images from the 'MSRC' dataset. Refer to the descriptions in Fig. 2 due to the space limitation.

(GSC) [16, 26], cooperative cuts (CC) [13, 27] and deep random field (DRF) [14, 28, 29] are most relevant to our method, and DRF is the state-of-the-art method. In these four methods, scribbles are used as interactions, while in our method rectangles are used. To highlight the improvement over GC, we also provide the results of GC with rectangles as interactions (denoted by R-GC).

For the unary potentials, we follow [4,12–14] to fit two Gaussian mixture models, each with five components, to the foreground and background seeds (user-specified or generated from the rectangles).

The pairwise potential incorporates an Ising prior and a contrast-dependent prior, with the penalty w_{ij} calculated by $w_{ij} = d_{ij}^{-1}(\beta_1 + \beta_2 \exp(-\|\mathbf{p}_i - \mathbf{p}_j\|_2^2/(2\sigma^2))$ [12–14], where $\beta_1$and β_2 are the relative weights for the two priors, σ^2 is the mean of squared gradients over $\mathcal{I}$ [4], d_{ij} is the spatial distance between neighboring pixels i and j, and $\|\cdot\|_2$ denotes ℓ_2-norm. For fair comparisons, we use the same experimental settings as those in GC, GSC, CC and DRF, namely, $\beta_1 = 2.5$, $\beta_2 = 47.5$, and an 8-neighbor graph structure.

Evaluation criteria. Here we use two commonly-used criteria:

1. Intersection-over-union score (**IoU**) [30]: the area of the intersection of the segmented object mask and the ground truth object mask divided by the area of their union.

2. Error rate (**Err.**) [13]: the number of incorrectly labeled pixels divided by the number of all the pixels in the image.

Qualitative comparisons. Qualitative results on five images from the 'FineStruct' dataset and six containing either FS or compact objects from the 'MSRC' dataset are shown in Fig. 2 and Fig. 3. As can be seen, Eg-GC produces complete and connected segmented objects with simpler interactions. Moreover, the visual differences demonstrate the advantage of Eg-GC over R-GC. Therefore, the effectiveness of the proposed interaction paradigm and the Eg-GC model is verified. In addition, the last three rows in Fig. 3 show that Eg-GC is also capable of handling compact objects.

Table 2. Interaction and running time averaged on all the images. Red is the best and blue the second best. Best viewed in color.

Meth.	GC	GSC	CC	DRF	R-GC	Eg-GC
Time1(s)	16.72	16.72	16.72	16.72	11.00	11.00
Time2(s)	0.63	1.09	4.92	16.03	1.37	2.39
Total(s)	17.35	17.81	21.64	32.75	12.37	13.39

Quantitative comparisons. Table 1 displays the quantitative comparisons of the exhibited results in Fig. 2 and the results *averaged on all the images* in the 'FineStruct' dataset. Note that the results of the 'MSRC' dataset are not included due to the absence of detailed ground truth. As can be seen, in most cases Eg-GC outperforms and it achieves the highest **IoU** and lowest **Err.** on average. Therefore, we can conclude that Eg-GC can effectively mitigate shrinking bias and handle FS objects. Furthermore, the improvement over R-GC verifies that the weighting adjustment does take effects.

Table 2 lists the averaged runtime of each method. Here **Time1** denotes the time for specifying enough interactions to yield satisfactory results. We collect the interaction time from five inexperienced users. For R-GC and Eg-GC, time for the iterative scheme to generate seeds (and further user editing if necessary) is also included for fair comparisons. **Time2** records the time for executing the entire algorithms except specifying interactions. Note that we have excluded the time-consuming K-means clustering of graph edges in CC and DRF (which costs about 28s) to highlight the time for inference. As the locations and sizes do not have to be strictly accurate, we see that drawing rectangles intuitively requires less time and manual efforts than scribbling, which saves nearly 6s on average. Due to the avoidance of NP-hardness and graph edge clustering, Eg-GC has lower demands of runtime in comparison to CC and DRF.

4. CONCLUSION

In this paper, we presented a novel method for fine-structured object segmentation within the graph cut framework. First, we developed a user-friendly and time-saving paradigm for simplifying the interactions. Interaction burden is thus considerably reduced, especially in the manually intractable fine structures. Second, the edge-guided graph cut model with weighting adjustment of graph edges is developed. It mitigates the shrinking bias of graph cut and further improves the completeness and connectivity of the segmented objects. Experimental results show that our method, *edge-guided graph cut model with interaction simplification*, achieves better or comparable performance in comparison to the state-of-the-art methods.

5. REFERENCES

[1] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[2] Yuri Boykov and Marie-Pierre Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *IEEE International Conference on Computer Vision*, 2001, pp. 105–112.

[3] Tony F. Chan and Luminita A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[4] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, ""GrabCut"–interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH*, 2004, pp. 309–314.

[5] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum, "Lazy snapping," in *ACM SIGGRAPH*, 2004, pp. 303–308.

[6] Leo Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[7] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov, "Grabcut in one cut," in *IEEE International Conference on Computer Vision*, 2013, pp. 1769–1776.

[8] Wallace Casaca, Luis Gustavo Nonato, and Gabriel Taubin, "Laplacian coordinates for seeded image segmentation," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2014, pp. 384–391.

[9] Yuri Boykov, Olga Veksler, and Ramin Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[10] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Extract maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society, Series B*, vol. 51, no. 2, pp. 271–279, 1989.

[11] Yuri Boykov and Vladimir Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[12] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother, "Graph cut based image segmentation with connectivity priors," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[13] Stefanie Jegelka and Jeff Bilmes, "Submodularity beyond submodular energies: Coupling edges in graph cuts," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1897–1904.

[14] Pushmeet Kohli, Anton Osokin, and Stefanie Jegelka, "A principled deep random field model for image segmentation," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1971–1978.

[15] Brian Price, Bryan Morse, and Scott Cohen, "Geodesic graph cut for interactive image segmentation," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3161–3168.

[16] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman, "Geodesic star convexity for interactive image segmentation," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3129–3136.

[17] Vladimir Kolmogorov and Yuri Boykov, "What metrics can be approximated by geo-cuts, or global optimization of length/area and flux," in *IEEE International Conference on Computer Vision*, 2005, pp. 564–571.

[18] Jan Stühmer, Peter Schröder, and Daniel Cremers, "Tree shape priors with connectivity constraints using convex relaxation on general graphs," in *IEEE International Conference on Computer Vision*, 2013, pp. 2336–2343.

[19] Vladimir Kolmogorov and Ramin Zabih, "What energy functions can be minimized via graph cuts?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.

[20] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.

[21] Mingkun Li and Ishwar K. Sethi, "Confidence-based active learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1251–1261, 2006.

[22] John Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[23] Stefanie Jegelka and Jeff Bilmes, "'Twigs&Legs' data set," `http://melodi.ee.washington.edu/~jegelka/cc/index.html`.

[24] John M. Winn, Antonio Criminisi, and Thomas P. Minka, "Object categorization by learned universal visual dictionary," in *IEEE International Conference on Computer Vision*, 2005, pp. 1800–1807.

[25] Vladimir Kolmogorov, "Max-flow/min-cut," `http://vision.csd.uwo.ca/code/`.

[26] Varun Gulshan, "Geodesic star convexity," `http://www.robots.ox.ac.uk/~vgg/research/iseg/`.

[27] Evan Shelhamer, "Cooperative cuts," `http://coopcut.berkeleyvision.org`.

[28] Pushmeet Kohli and Philip H. S. Torr, "Effciently solving dynamic markov random fields using graph cuts," in *IEEE International Conference on Computer Vision*, 2005, pp. 922–929.

[29] Anton Osokin, "Code for the algorithms of cooperative cuts," `http://research.microsoft.com/en-us/um/people/pkohli/code/coopCuts_CVPR2013.zip`.

[30] PASCAL Visual Object Classes Challenge 2009 (VOC2009), ," `http://host.robots.ox.ac.uk/pascal/VOC/voc2009/`.