

Interactive image segmentation based on the appearance model and orientation energy

Shaojun Qu ^a, Huang Tan ^a, Qiaoliang Li ^{a,*}, Zili Peng ^b

^a Hunan Normal University, 36 Lushan Rd, Yuelu District, Changsha, 410081, China

^b Hunan Youmei Technology Development Co., Ltd., Chuanggu Industrial Park, No. 568 Queyuan Rd, Tianxin District, Changsha, China



ARTICLE INFO

Communicated by Nikos Paragios

Keywords:

Graph cut
Appearance model
Orientation energy
Pseudoflow
Interactive image segmentation
Binary segmentation

ABSTRACT

Tang et al. (2013) proposed a graph-based image segmentation model by minimizing the distance between the object and background appearance overlap models. This model is very effective for interactive image segmentation. However, it is prone to isolated nodes when the colors or other appearances characteristic of the object and background are very similar. To improve the performance of this algorithm and related algorithms, we add new spatial distance and contour orientation energy terms to the energy function. Accordingly, we modify the construction of the energy graph. We add terminal nodes S and T and add prior constraints. Finally, we use the pseudoflow algorithm proposed by Hochbaum to calculate the maximum flow of the new energy graph. A large number of experiments on the MSRA dataset, BSD dataset and GrabCut dataset show that the results of the proposed method are better than those of many recently proposed image segmentation methods. The code is available at <https://github.com/powerhope/AMOE>.

1. Introduction

Due to its effective segmentation properties, the image segmentation method based on graph cuts has become a popular method in the image segmentation field in the last decade. The essence of the method is to transform an image segmentation problem into an energy cost function minimization problem. There are many studies on image segmentation based on graph cuts (Greig et al., 1989; Wu and Leahy, 1993; Cox et al., 1996; Ding et al., 2001; Boykov et al., 2001; Boykov and Kolmogorov, 2004; Rother et al., 2004; Tang et al., 2013). A comprehensive review of the research (Peng et al., 2013) is provided in the literature. However, one common drawback of these methods is that the energy defined in these paper on image segmentation with graph cuts is prone to producing isolated regions. An example is the energy function defined by Boykov et al. (2001) for image segmentation. This model is prone to generating isolated regions, as discussed in previous work (Wu and Leahy, 1993; Shi and Malik, 2000).

GrabCut (Rother et al., 2004) is a very popular image segmentation method that iteratively uses the Gaussian mixture model (GMM) to evaluate the appearance model from a given bounding box. Thus, the GMM is slow. To improve the efficiency of GrabCut, Tang et al. (2013) proposed a new method, named OneCut, by replacing the log-likelihood with a new energy term to explicitly measure the L_1 distance between the appearance models of the object and background. This problem can be globally optimized with graph cuts only once. Thus, it effectively reduces the computational complexity and maintains good

segmentation results. However, for some images, this method is prone to producing isolated regions for the following reasons: first, the energy defined in OneCut (Tang et al., 2013) for image segmentation with graph cuts is prone to producing isolated regions. The reason why OneCut is prone to isolated regions is that the value of the minimum cut increases as the number of edges connecting the two regions increases. Assuming that the weight of an edge is inversely proportional to the distance between the two nodes, when the nodes are separately divided, the resulting cut value will be very small. The second reason is that OneCut uses the L_1 distance; this distance considers only the similarity between pixels and ignores the spatial distance between pixels. A large number of experiments show that the segmentation results are not good when the appearance difference between the object and background is low. Finally, OneCut is very sensitive to seed points: for some images, many seed points are needed to obtain better segmentation results, and for other images, when the positions of the marked seed points are changed slightly, the segmentation results are very different.

In this paper, we develop an improved interactive image segmentation algorithm. We present a new energy function that considers the spatial distance and contour orientation energy between pixels and the brightness cue based on the intervening contours. The new method can obtain good segmentation results and overcome the problems of isolated regions caused by appearance similarity in OneCut. Then, to minimize the proposed energy function, we modify the construction of

* Corresponding author.

E-mail address: liqiaoliang@hunnu.edu.cn (Q. Li).

the energy graph, and then, we use the pseudoflow algorithm (abbreviated HPF) (Hochbaum, 2008) to find the minimum cuts of the graph. The method improves the segmentation accuracy. In addition, the user can further modify the segment by adding markers after obtaining the initial segmentation results. Thus, the algorithm is more suitable for some practical applications. Although some extra computation is needed for the new boundary term in the energy function, the time complexity of the HPF algorithm is $O(mn \log n)$, where n is the number of nodes and m is the number of arcs, making it a strong polynomial algorithm (Hochbaum, 2008). Experimental results show that the computational time of the proposed algorithm is very effective. For the application scenario, our algorithm, similar to most other interactive image segmentation methods, plays a pivotal role in many computer vision applications, for example, image editing, intelligent transportation, medical image analysis and processing, and unmanned driving (Peng et al., 2019).

The rest of the paper is structured as follows. Section 2 reviews the relevant work. Section 3 describes the proposed method. Section 4 presents the experimental results and comparisons. Finally, we conclude in Section 5.

2. Related work

In 1989, Greig et al. (1989) first proposed the use of graph cuts to solve the problem of binary image restoration by solving an energy optimization problem. Since it was confined to the problem of binary image processing, it did not attract people's attention at the time. Later, Wu and Leahy (1993) proposed segmenting an image by finding the minimum of the global cost function with the minimum cut of the associated undirected graph. The basic idea is to map the pixels of the image to the vertices of the graph. The similarity measurement between two pixels corresponds to the weight of the edge between two vertices in the graph. In this way, the segmentation of images is transformed into the minimum cut of weighted graphs. Boykov (Boykov et al., 2001; Boykov and Kolmogorov, 2004) proposed an interactive graph cut method that also turns image segmentation into graph theory and associates it with the minimum cut. The minimum cut problem can usually be solved by finding the maximum flow from the source s to the sink t ; that is, the minimum cut problem and the maximum flow problem are equivalent. Rother et al. (2004) enriched the graph cut method and developed a more powerful iterative version of the segmentation algorithm, GrabCut, by replacing the Gaussian mixture model with a histogram model and iterative estimation. Based on GrabCut, Tang et al. (2013) proposed a new energy term, replacing the data likelihood term with the submodular L_1 distance. The new energy term explicitly measures the L_1 distance of the appearance models of the object and background, and the term can be optimized by the graph cut method, which does not require iterative use of graph cuts in the computation. Thus, it effectively reduces the computational complexity and maintains good segmentation results. Later, Tang et al. (2014) proposed a parametric maximum flow algorithm. Ren et al. (2017) presented a more comprehensive review of the GrabCut algorithm, and they solved the segmentation degradation problem when the number of superpixels was low. To avoid an unbalanced small point set in the minimum cut, Shi and Malik (2000) proposed to use normalized cuts to minimize the disassociation between regions while maximizing the association with the same region. They approximated normalized cuts by generalized eigenvectors. Chew and Cahill (2015) reformed Ncut to handle the must-link constraints and the cannot-link constraints in a soft-constrained manner, enabling the user to adjust the degree of constraint satisfaction. Qu et al. (2019) proposed supervised image segmentation based on superpixels and improved normalized cuts.

Another related graph-based interactive image segmentation algorithm is the random walk algorithm (Grady, 2006), which calculates the probability of each nonseed point to the seed points in the image. Since then, many algorithms based on random walk have been

proposed. Dong et al. proposed a unified sub-Markov random walk algorithm (Dong et al., 2016), which is considered to be an extension of lazy random walk (LRW), random walk with restart (RWR) and partially absorbing random walk (PARW). By developing and defining the diffusion process on any graph, Bampis proposed a graph-driven graph cut method (Bampis et al., 2017). A normalized random walk algorithm and LRW variants were developed by correlating popular infection-recovery propagation models with random walk algorithms.

He et al. (2020) proposed an improved GrabCut algorithm that combines segmentation and multiscale feature extraction into a unified model. Tang et al. (2018) used the normalized cut as the loss function of weakly supervised CNN segmentation, which effectively realized the dense Gaussian kernel function in linear time by fast bilateral filtering. Their normalized cut loss segmentation method made the quality of weakly supervised training significantly close to that of fully supervised training. Shuai et al. (2018) proposed a directed acyclic graph-recurrent neural network (DAG-RNN) for context aggregation of feature maps of local connections. The DAG-RNN was placed on top of a pre-trained convolutional neural network (CNN) to embed the context into local features to enhance the representation of local features. Lateef and Ruichek (2019) comprehensively summarized and analyzed studies on semantic segmentation of deep learning in recent years.

The minimum s-t cut is a classic combinatorial optimization problem. It is also an important building module in many visual and imaging algorithms. Hochbaum introduced the pseudoflow algorithm in Chandran and Hochbaum (2009), Fishbain et al. (2016) and applied it to solve the problem of minimum s-t cut. A theoretical comparison and experimental verification were carried out in terms of the execution time and memory utilization of the push-relabel, augmenting path and partial augment-relabel algorithms. Experimental results showed that the pseudoflow algorithm is more efficient and uses less memory than the other three algorithms. It is a good choice for many real-time computer vision problems that require solving the problem of the minimum s-t cut.

2.1. Graph cut

Boykov et al. (2001) proposed applying graph cut technology to image segmentation. Such methods associate the image segmentation problem with the graph minimum cut problem. First, an image is represented by a graph $G = \langle V, A \rangle$, where V and A are the sets of vertices and edges, respectively. The graph adds two vertices based on the ordinary graph. These two vertices are represented by the symbols "S" and "T" and they are collectively called terminal vertices. All other vertices are connected to these two vertices to form part of the edge set. Thus, there are two kinds of vertices and two kinds of edges (t-link: terminal link; n-link: neighbor link) in the graph cuts.

The energy function of an image is represented by $E(P) = \alpha R(P) + B(P)$, where P defines a segment, $R(P)$ is the regional term, $B(P)$ is the boundary item, and α is the importance of regional and boundary terms. The cut is defined as follows:

$$\text{cut}(P, \bar{P}) = \sum_{i \in P, j \in \bar{P}} w(i, j), \quad (1)$$

where P and \bar{P} indicate that graph G is divided into two disjoint regions and $w(i, j)$ is the edge weight between pixels i and j . The minimum cut is the cut in which the sum of the edge weights is the smallest among all cuts in the graph. The minimum cut can be obtained by solving the maximum flow.

2.2. GrabCut

Rother et al. (2004) proposed the GrabCut algorithm, which replaced the grayscale histogram with a GMM of RGB color images. Then, the graph cut algorithm is iteratively used to estimate the GMM

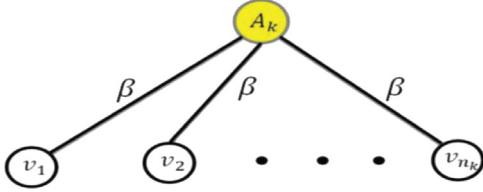


Fig. 1. Graph construction for E_{L_1} : nodes v_1, v_2, \dots, v_{n_k} corresponding to the pixels in bin k are connected to the auxiliary node A_k using undirected links. The capacity of these links is the weight of appearance overlap term $\beta > 0$.

parameters until the entire algorithm converges. The object energy function can be expressed as follows:

$$E(S, \theta^0, \theta^1) = - \sum_{p \in \Omega} \ln Pr(I_p | \theta^{S_p}) + |\partial S|. \quad (2)$$

The first term is the data term, and the second is the smoothness term. Ω is the image itself, and P is the pixels. I_p indicates that pixel P is labeled I_p . The boundary term $|\partial S| = \sum W_{pq} |S_p - S_q|$, with $W_{pq} = \frac{1}{\|p - q\|} \cdot \exp[-\frac{\Delta I^2}{2\delta^2}]$, where $s_p \in \{0, 1\}$ are the binary indicator variables for segment $S \subset \Omega$. Initial appearance models θ^0 and θ^1 are computed from a given bounding box. In GrabCut, it is assumed that the object and the background follow a GMM whose parameters θ^0 and θ^1 is the log-likelihood based on a GMM and cannot be maximized by graph cuts. In the iteration process, θ^0 and θ^1 are obtained by the expectation maximization algorithm (EM), and then, the graph cut method is applied to minimize the energy function and obtain a segment. GrabCut repeatedly iterates these two processes so that θ^0 and θ^1 gradually approach the true value. Thus, the computation cost of GrabCut is very large, and the algorithm is slow.

2.3. GrabCut in one cut

GrabCut in one cut (Tang et al., 2013) is based on the objective function of GrabCut, and the Gaussian mixture model is replaced by the L_1 distance between the object and background appearance models:

$$E(S) = -\|\theta^S - \theta^{\bar{S}}\|_{L_1} + |\partial S|. \quad (3)$$

Here, the term L_1 actually calculates the L_1 difference between the non-normalized histograms of S and \bar{S} , and the L_1 -norm utilized is better than any other form of approximate appearance overlap terms regarding computation time. It is a high-order energy term and a submodular function (Lovász, 1983) that can be solved by the graph cut method. Compared with GrabCut, this energy function does not need to use iterations for optimization. OneCut obtains the result by using the graph cut method only once. Therefore, the segmentation efficiency is improved.

Term L_1 in $E(S)$ is optimized with a graph cut by adding auxiliary nodes. k auxiliary nodes A_1, A_2, \dots, A_k are added according to the number of bin values k . Then, all the pixels of the k th bin are connected to the new k th auxiliary node A_k , and the weight of these connections is set to $\beta = 1$, as shown in Fig. 1.

There are two ways to implement OneCut. One is similar to that for GrabCut. First, the object to be segmented is selected with a bounding box. The pixels outside the bounding box are used as the background. The segmentation energy function $E(S)$ is expressed as follows:

$$E(S) = |\bar{S} \cap R| - \beta \|\theta^S - \theta^{\bar{S}}\|_{L_1} + \lambda |\partial S|. \quad (4)$$

$R \subset \Omega$ indicates a binary mask that is consistent with the bounding box, S indicates a segment, the first term is standard dilation in the bounding box R , the second term is the appearance overlap penalty, and the last term is the smoothing term, $|\partial S| = \sum W_{pq} |S_p - S_q|$, with $W_{pq} = (1/\|p - q\|) \cdot \exp[-\Delta I^2/2\delta^2]$. $s_p \in \{0, 1\}$ are the binary indicator

variables for segment $S \subset \Omega$, ΔI is the color difference between pixel p and q , and δ^2 is set to be the mean of ΔI^2 over the image.

β is the penalty factor of the approximate appearance overlap, defined as follows:

$$\beta = \frac{|R|}{-\|\theta^R - \theta^{\bar{R}}\|_{L_1} + |\Omega|/2} \cdot \beta'. \quad (5)$$

Here, β' is a tuned global parameter.

The second way to implement OneCut is to mark the object and the background seeds by the method in Boykov et al. (2001). The segmentation energy function is expressed as follows:

$$E(S) = -\beta \|\theta^S - \theta^{\bar{S}}\|_{L_1} + \lambda |\partial S|. \quad (6)$$

One shortcoming of OneCut is that it is prone to segment isolated nodes when the colors or other appearances of the object and background are very similar.

3. Proposed method

To overcome the shortcomings of OneCut, we add a new boundary term to the energy function in this paper and consider the spatial distance between pixels and orientation energy (Shi and Malik, 2000; Leung and Malik, 1998), which effectively reduces the isolated nodes. The cost function is defined as follows:

$$E(S) = |\bar{S} \cap R| - \beta \|\theta^S - \theta^{\bar{S}}\|_{L_1} + \lambda |\partial S| + \gamma |\partial OE|. \quad (7)$$

The first term in $E(S)$ has the same meaning as in Eq. (4). The second and third terms in $E(S)$ have the same meaning as in Eq. (5). γ can adjust the weight of the orientation energy.

$$|\partial OE| = B(\Omega \cup Aux). \quad (8)$$

Here,

$$|\partial S| = \sum W_{pq} |S_p - S_q|, \\ W_{pq} = \frac{1}{\|p - q\|} \cdot \exp[-\frac{\Delta I^2}{2\delta^2}], \quad (9)$$

where $s_p \in \{0, 1\}$ are the binary indicator variables for segment $S \subset \Omega$, ΔI is the color difference between pixel p and q , and δ^2 is set to be the mean of ΔI^2 over the image. Aux is the new set of auxiliary nodes, and

$$B(\Omega \cup Aux) = \sum_{i, j \in \Omega \cup Aux} B(i, j). \quad (10)$$

$$B(i, j) = W_s(i, j) \times W_c(i, j). \quad (11)$$

$$W_s(i, j) = \begin{cases} \exp(-\frac{\|D(i) - D(j)\|_2^2}{\sigma_D^2}) & \text{if } \|D(i) - D(j)\|_2 < r, \\ 0 & \text{if } \|D(i) - D(j)\|_2 \geq r, \\ 0 & \text{if } i \text{ or } j \in Aux, \end{cases} \quad (12)$$

where $D(i)$ and $D(j)$ denote the pixel spatial location; this information regarding connecting adjacent pixels is very useful. σ_D can be estimated as camera noise. r is a constant.

$W_c(i, j)$ estimates the similarity between pixel i and pixel j by measuring the intervening contour (Leung and Malik, 1998) between pixels i and j . We first define the oriented energy before defining $W_c(i, j)$.

Let $F_1(x, y)$ be the second derivative of an elongated Gaussian kernel and $F_2(x, y)$ be the Hilbert transform of $F_1(x, y)$ (Leung and Malik, 1998).

$$F_1(x, y) = \frac{d^2}{dy^2} \left(\frac{1}{C} \exp\left(\frac{y^2}{\sigma^2}\right) \exp\left(-\frac{x^2}{\lambda_1^2 \sigma^2}\right) \right), \quad (13)$$

$$F_2(x, y) = \text{Hilbert}(F_1(x, y)), \quad (14)$$

where σ is the scale and λ_1 is the elongation of the filter. C is a constant. The orientation energy at angle 0° is defined as follows (Leung and Malik, 1998):

$$OE_{0^\circ} = (I * F_1)^2 + (I * F_2)^2, \quad (15)$$

where I is the image before convolution. OE_{0° has a maximum response for horizontal contours. At each pixel, we can define the orientation energy and the orientation as follows (Leung and Malik, 1998):

$$OE_{con}(x, y) = \max_{\phi} OE_{\phi}(x, y). \quad (16)$$

$$\phi(x, y) = \arg \max_{\phi} OE_{\phi}(x, y). \quad (17)$$

Leung and Malik (1998) note that the orientation energy has four advantages. From the orientation energy, we can compute the dissimilarity between two pixels due to the curvilinear continuity factor. Intuitively, two pixels belong to two different groups if there is a contour separating them. The dissimilarity is stronger if the contour separating them is extended. Formally, the dissimilarity is defined as follows (Leung and Malik, 1998):

$$d_{edg}(i, j) = OE_{con}(\hat{x}) - \text{avg}(OE_{con}(i), OE_{con}(j); \hat{x}), \quad (18)$$

where $\hat{x} = \arg \max_{x \in l} OE_{con}(x)$. l is the straight line between i and j ; \hat{x} is the location where the orientation energy is maximum on l ; and $\text{avg}(OE_{con}(i), OE_{con}(j); \hat{x})$ is the weighted average orientation energy of i and j . We combine contour information with intensity and/or color to produce the overall similarity measure between pixels i and j as follows:

$$W_c(i, j) = \exp(-\alpha_{edg} d_{edg}(i, j)). \quad (19)$$

α_{edg} is the weighting coefficient.

From $W_c(i, j)$, we can compute the dissimilarity between two pixels due to the curvilinear continuity factor.

According to OneCut and Leung and Malik (1998), our objective energy function is submodular and can be minimized by utilizing the efficient maximum flow algorithm.

Eq. (11) introduces the orientation energy and the distance between pixels. There are three effects of this: It (1) helps remove small isolated regions, as shown in Fig. 4; (2) solves the mis-segmentation problem when the object and background colors are similar, as shown in Fig. 5; and (3) reduce mis-segmentation due to equal bin values of the object and background, as shown in Figs. 5 and 6.

In addition, we improved the construction of the graph method in this paper and finally used the HPF (Hochbaum, 2008) algorithm to minimize energy. The proposed method not only avoids isolated nodes but also reduces the sensitivity to the marked seeds and improves the segmentation accuracy. The graph is constructed as follows:

Take graph $G = (V, A)$, where V is the set of vertices with n (image width \times height) nodes of the image (a pixel is a vertex), new auxiliary nodes k and terminal nodes S and T , A is the set of edges, and there are three types of edges. The first type of edge is the adjacent edges (n -links) between the image pixels, which are denoted N , and we adopt 8 adjacency. The second type of edge links between k th auxiliary node A_k (which are the same as those in OneCut) and all the pixels that belong to the k th bin, which is denoted U . The third type of edge is those with one end in S and the other end in the object seed set and those with one end in T and the other end in the background seed set. Fig. 2 shows a schematic diagram of nine pixels. The blue vertices are new auxiliary nodes of the L_1 term. The red edges are the n -links. The black edges are the edges from the pixels of the k th bin to the new k th auxiliary node A_k . The red vertex is the object seed, and the yellow vertex is the background seed.

The difference from OneCut lies in the graph construction method and the edge weight calculation. In the graph, we add terminal nodes S and T and add prior constraints.

Table 1 gives the weights of the edges, where p and q are any two vertices.

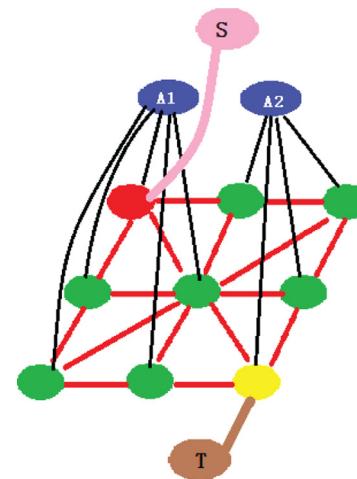


Fig. 2. Mapping example of 9 pixels. The blue vertices are new auxiliary nodes. The red edges are the n -links, and the black edges are the edges from the pixels of the k th bin to the new k th auxiliary node A_k . The red vertex is the object seed, and the yellow vertex is the background seed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Calculation of edge weights.

Edge	Weight	For
$\{p, q\}$	$W_{pq} + B(p, q)$	$\{p, q\} \in N$
$\{p, q\}$	β	$p \in N, q \in \text{Aux}$
$\{p, S\}$	$INFTY = 100\,000\,000$	$p \in Obj$
$\{p, T\}$	$INFTY = 100\,000\,000$	$p \in Background$

After graph G is established, we use the HPF algorithm (Hochbaum, 2008; Chandran and Hochbaum, 2009; Fishbain et al., 2016) provided by the author to obtain the optimal segment. Chandran and Hochbaum (2009) compared the performance of HPF, in terms of execution time and memory utilization, with that of three leading published algorithms: (1) Goldberg's and Tarjan's push-relabel; (2) Boykov's and Kolmogorov's augmenting path; and (3) Goldberg's partial augment-relabel algorithms. The results demonstrate that, in general, the HPF algorithm is more efficient and utilizes less memory than these three algorithms.

The improved algorithm can segment any specified object in the image, and the user can further enhance segmentation by adding markers after the initial segmentation result is obtained. The algorithm flowchart is shown in Fig. 3. First, the image is selected. Second, a rectangular box is used to select the object of interest. Third, the image is scribbled, where green/blue scribbles specify the object/background regions. Fourth, the related weights are calculated, and a graph is constructed. Fifth, the graph optimization algorithm is run, and segmentation results are obtained. If the user is not satisfied with the segmentation result, then he or she can repeat the third to fifth steps until a satisfactory result is obtained. The last line remaining shows the markers added to a position with segmentation error (the background between the two legs). After adding background seed points, we obtain satisfactory segmentation results.

The image segmentation algorithm framework proposed in this paper is shown in algorithm 1.

4. Experiments and comparisons

To verify the effectiveness of the algorithm, we prepared 2 image datasets. First, we tested on the MSRA dataset (Liu et al., 2007) and

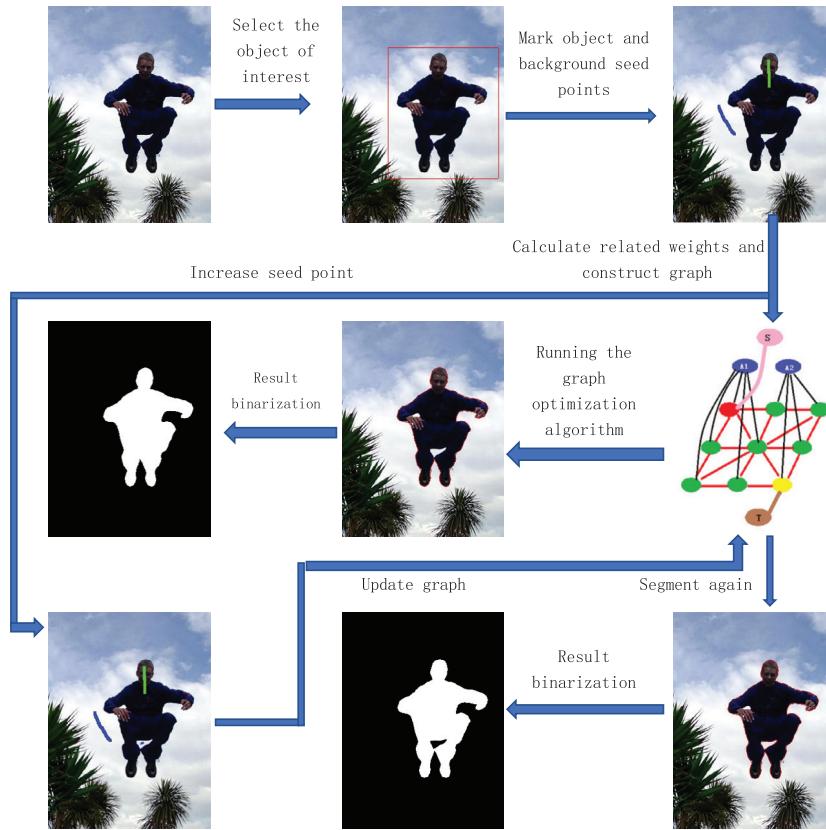


Fig. 3. Flowchart of the algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

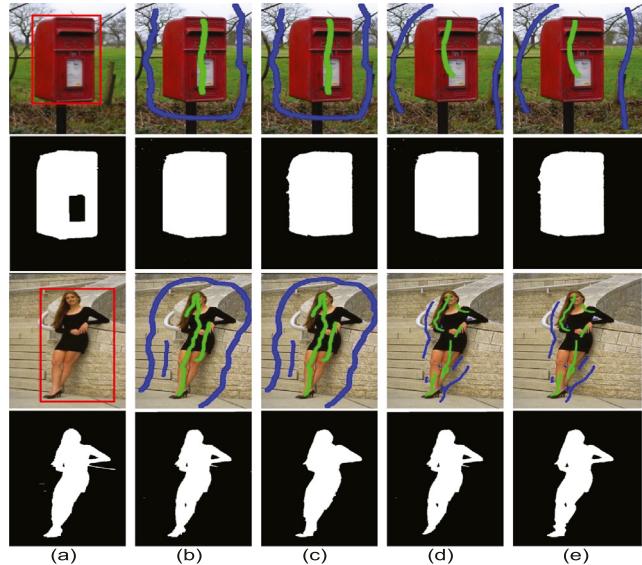


Fig. 4. Comparison of our method with OneCut. The first and third rows are user input, and the second and fourth rows are segmented results. (a) OneCut with a box. (b) OneCut with seeds. (c) Our method (the input seed points are the same as those in OneCut). (d) OneCut (the input seed points are the same as those in our method). (e) Our method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

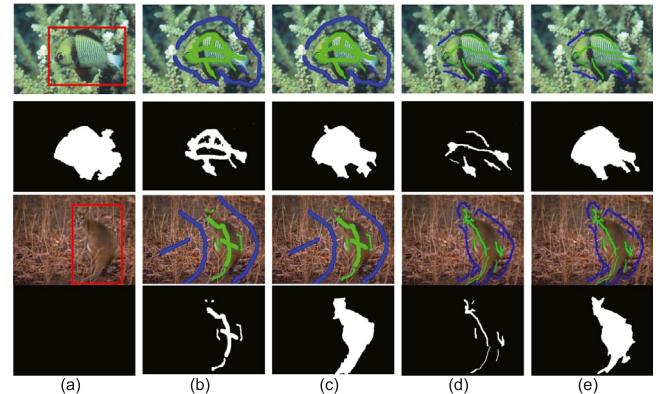


Fig. 5. Comparison of our method with OneCut. The first and third rows are user input, and the second and fourth rows are segmented results. (a) OneCut with a box. (b) OneCut with seeds. (c) Our method (the input seed points are the same as those in OneCut). (d) OneCut (the input seed points are the same as those in our method). (e) Our method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the BSD dataset (Martin et al., 2001). We randomly selected 250 images, including images with a single background and less texture, images with a more complex background and more texture, and images whose foreground and background colors were very similar. Image types include animals, buildings, people, fruit, airplanes, flowers, cars, plants, etc. The binary ground truth of MSRA is provided by the dataset, and the ground truth of BSD is provided by McGuinness and O'Connor

Algorithm 1 Interactive Image Segmentation Based on the Appearance Model and Orientation Energy (AMOE)

Input:

im: the original color image.

box: the object of interest.

ObjectSeeds: the green scribbles specify the object seeds.

BackgroundSeeds: the blue scribbles specify the background seeds.

Parameter settings: $\beta' = 0.9$, $\lambda = 9.0$, $\gamma = 1$, $r = 10$, $bin = 64^3$, $\alpha_{edg} = 1$, and use 8 adjacency for the pixels in the image.

Output:

seglabel: mask image.

imgseg: segmentation result.

Method:

Step 1. Read the original image (*im*);

Step 2. Select the object of interest with a rectangle (*box*);

Step 3. Scribble the image, where the green/blue scribbles specify the object (*ObjectSeeds*)/background (*BackgroundSeeds*) seeds;

Step 4. Compute the term $|\bar{S} \cap R|$ according to the *box* selected in step 2;

Step 5. Compute the term $\|\theta^S - \theta^{\bar{S}}\|_{L_1}$;

Step 6. Compute β according to Eq. (5);

Step 7. Compute smooth term $|\partial S|$ according to Eq. (9);

Step 8. Compute $W_S(i, j)$ according to Eq. (12);

Step 9. Compute orientation energy $W_c(i, j)$ according to Eq. (19);

Step 10. Construct graph $G = (V, A)$ according to Table 1;

Step 11. Use the pseudoflow algorithm to optimize the energy function, and obtain the segmentation result of the graph;

Step 12. Output preliminary segmentation results;

Step 13. If the segmentation result is unsatisfactory, then seed points can be modified or added, and the segmentation result can be further optimized (repeat steps 3–12);

Step 14. Obtain the final satisfactory result.

(2011). Second, we tested on the GrabCut dataset. Markers for GrabCut were obtained from Perret et al. (2015).

The evaluation parameters are based on the Jaccard similarity coefficient (Jac) (Qu et al., 2019; Jain and Dubes, 1988), precision, recall, F-measure and mean error rate (ME).

We ran all methods on a computer with an Intel Core (TM) CPU i7-4710MQ 2.50 GHz and 8 GB RAM system.

4.1. Evaluation index

4.1.1. Jaccard similarity coefficient

For binary segmentation, Jac is defined by Jain and Dubes (1988).

$$Jac(A, A') = \frac{|A \cap A'|}{|A \cup A'|}, \quad (20)$$

where *A* is the ground truth and *A'* is the segmentation result.

4.1.2. Average precision, recall, F-measure, and ME

To objectively evaluate our method, we further evaluate the precision, recall, F-measure and ME. The average precision, recall, F-measure, and ME are evaluated on the entire ground truth datasets, with the F-measure defined as the harmonic mean of the precision and recall:

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}. \quad (21)$$

The F-measure is used to determine the importance of precision for recall. The general setting is $\beta^2 = 0.3$ so that the precision is slightly higher than the recall rate.

Given two point sets *A* and *A'*, the ME is defined as follows:

$$ME = 1 - \frac{n_1 + n_2}{row \times col}. \quad (22)$$

Table 2

Comparison of the results of our method and OneCut with Jac.

Method	Avg.	Min.	Max.
OneCut(16^3 bins)	0.4348	0.0101	0.9801
OneCut(32^3 bins)	0.5521	0.0114	0.9867
OneCut(64^3 bins)	0.5952	0.0114	0.9863
OneCut(128^3 bins)	0.6383	0.0118	0.9869
OneCut(256^3 bins)	0.6776	0.0122	0.9871
Our(16^3 bins)	0.8964	0.7264	0.9835
Our(32^3 bins)	0.8956	0.7281	0.9835
Our(64^3 bins)	0.8971	0.7282	0.9835
Our(128^3 bins)	0.8958	0.7284	0.9834
Our(256^3 bins)	0.8955	0.7295	0.9834

n_1 and n_2 represent the number of pixels belonging to the foreground and background in *A* and *A'*, respectively. *row* is the width of the image, and *col* is the height of the image (Qu et al., 2019).

4.2. Setting related parameters

To understand the influence of the number of color bins on the experimental results, we conduct a comparative experiment on the GrabCut dataset. The seed points of the images are from Perret et al. (2015) (<https://perso.esiee.fr/~perretb/markerdb/>). These authors marked several seed sets of different qualities, and we selected the seeds “Marker_sk-sk.png” with fewer points as the seed points for segmentation.

In the experiment, the number of bins has the greatest impact on the results of OneCut segmentation. Therefore, we fix the values of the three parameters β' , λ and r at 0.9, 9, and 10, respectively. Then, the number of bins is set at 16, 32, 64, 128, and 256. The relationship between pixels adopts 8 adjacency. In our proposed method, $\gamma = 1$, and $\alpha_{edg} = 1$. The results are shown in Table 2.

A larger Jaccard index is better. As we increase the number of color bins, the Jac for the OneCut method increases. With 256 bins, the OneCut method is the most accurate. The Jaccard index has a relatively large change. This is because with more bins, more auxiliary nodes are used, but each auxiliary node is connected to fewer pixels. The connectivity density decreases. It is important to see the effect of working with a larger number of bins, as some objects might only be distinguishable from the background when using a larger dynamic range. The experimental results show that our method is very slightly affected by the different bin values.

In subsequent experiments, for the OneCut method, the bin value is set to 256^3 , and the bin value in our method is set to 64^3 .

4.3. Analysis of OneCut shortcomings

Through experiments, we found the shortcomings of OneCut: First, energy functions (4) and (5) are prone to segment isolated nodes. There are some isolated points in columns (a), (b) and (d) in Fig. 4 and columns (b) and (d) in Fig. 5.

OneCut is prone to isolated nodes because OneCut uses the L_1 distance between the appearance models of the background and object, so only the color similarity between pixels is considered and the spatial distance between pixels, i.e., adjacency, is weakened.

Second, it is easy to cause mis-segmentation when there are similar colors in the object and the background and when the contrast between the object and the background is low or the texture is complex. As shown in Fig. 5, there are some pixels with similar colors in the object and the background of the image in the first row, and the segmentation results have many errors in columns (a), (b) and (d). The object and background of the image in the first and third rows have more similar colors, and the object is not segmented in columns (a), (b) and (d). The reason for this phenomenon is that pixels of similar colors have very close bin values in the calculation.

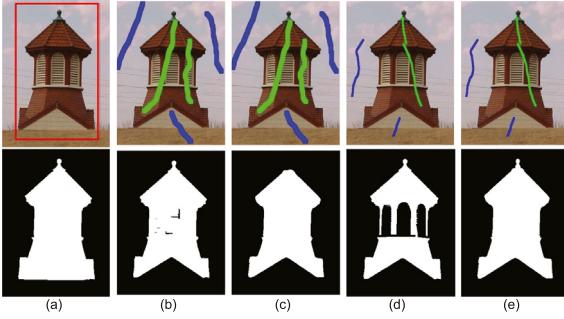


Fig. 6. Comparison of our method with OneCut. The first row is user input, and the second row is segmented results. (a) OneCut with a box. (b) OneCut with seeds. (c) Our method (the input seed points are the same as those in OneCut). (d) OneCut (the input seed points are the same as those in our method). (e) Our method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3

Performance comparison of six typical image segmentation methods.

Method	Jac		
	Avg.	Min.	Max.
Our	0.9242	0.6617	0.9923
OneCut	0.8634	0.2518	0.9960
pPBC-KL	0.9083	0.1778	0.9974
pPBC-BHA	0.7783	0.0485	0.9968
pPBC	0.7925	0.0416	0.9973
SSNCUT	0.8519	0.0249	0.9927

Table 4

Exchange of markers for four method.

Method	Seeds	Jac		
		Avg.	Min.	Max.
OneCut	SSNCUT	0.6980	0.0254	0.9964
OneCut	Our	0.7665	0.0407	0.9964
SSNCUT	OneCut	0.5341	0.0	0.9927
SSNCUT	Our	0.4337	0.0	0.9927
Our	OneCut	0.8662	0.3681	0.9914
Our	SSNCUT	0.8366	0.3681	0.9914

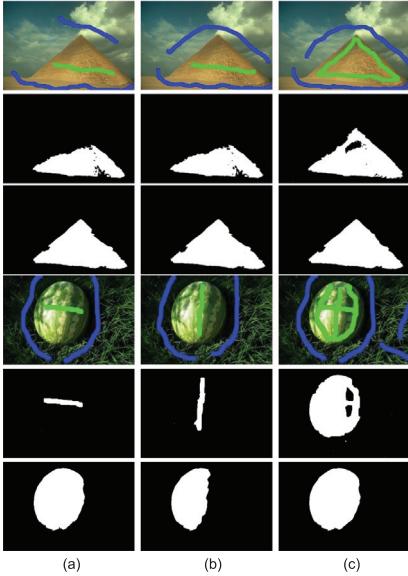


Fig. 7. Effect of user input position on segmentation results with OneCut. The first and fourth rows are different user inputs. Lines 2 and 5 are the segmentation results of OneCut. Lines 3 and 6 are our segmentation results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5

Performance comparison of five typical image segmentation methods.

Method	Jac		
	Avg.	Min.	Max.
OneCut	0.6776	0.0122	0.9871
RW	0.8860	0.6941	0.9865
pPBC-KL	0.6587	0.2652	0.9383
pPBC-BHA	0.6587	0.2652	0.9383
pPBC	0.8194	0.3855	0.9867
SSNCUT	0.5846	0.0746	0.9821
Our	0.8971	0.7282	0.9835

For OneCut, PBO (pPBC-KL, pPBC-BHA and pPBC) and SSNCUT, we use the code provided by the authors for optimal segmentation, and all parameters use the values suggested in the paper. We first tested each algorithm on 250 images.

Similar to most interactive image segmentation methods, we need to manually select the appropriate foreground and background seed points. The number and location of seed points have different effects on each segmentation method. The partial segmentation method is very sensitive to the number and location of seed points. In the experiment, no marker can make the segmentation results of all methods reach optimal or close to optimal results. To achieve optimal labeling for each method, we marked seed points separately for each method (Qu et al., 2019). We marked four sets of seeds for our method, OneCut, SSNCUT, and PBO (pPBC-KL, pPBC-BHA, and pPBC use the same seed point set). The first 3 sets of seeds were marked with a brush, and the last set of seed points was marked with a rectangular bounding box.

In our proposed method, where the parameter is the same as that in OneCut, the value settings are consistent with that in OneCut. We take $\beta' = 0.9$, $\lambda = 9.0$, and $r = 10$ and use 8 adjacency for the pixels. For our method, $\gamma = 1$, and $\alpha_{edg} = 1$. For the OneCut method, the number of bins is set to 256^3 , and for our method, the bin number is set to 64^3 .

The results are shown in Fig. 8 and Table 3. OneCut is the result obtained by using the cost function (6). The greater the Jaccard index is, the higher the segmentation accuracy.

In Table 3, we compared the average, minimum and maximum Jaccard indexes of each method.

Part of the segmentation results are shown in Fig. 9. In the pPBC-BHA method, there are many segmentation errors in the second, sixth,

Third, when the pixels of the object and the background have significant differences in colors, some bin values may also be very similar, e.g., the bottom of buildings in Fig. 6(a) and the windows in Fig. 6(b) and (d), which results in mis-segmentation.

Fourth, OneCut is very sensitive to the marked seed pixels. In addition, many images need many marked pixels to obtain better segmentation results. Therefore, a thicker marker is used in OneCut. As shown in Fig. 7, the segmentation results are very different when the marker positions are different.

4.4. Evaluation of the Jaccard similarity coefficient

We compared our method with five other methods, i.e., OneCut, three pseudobound optimization (PBO) variants (Tang et al., 2014) and semisupervised normalized cuts (SSNCUT) (Chew and Cahill, 2015).

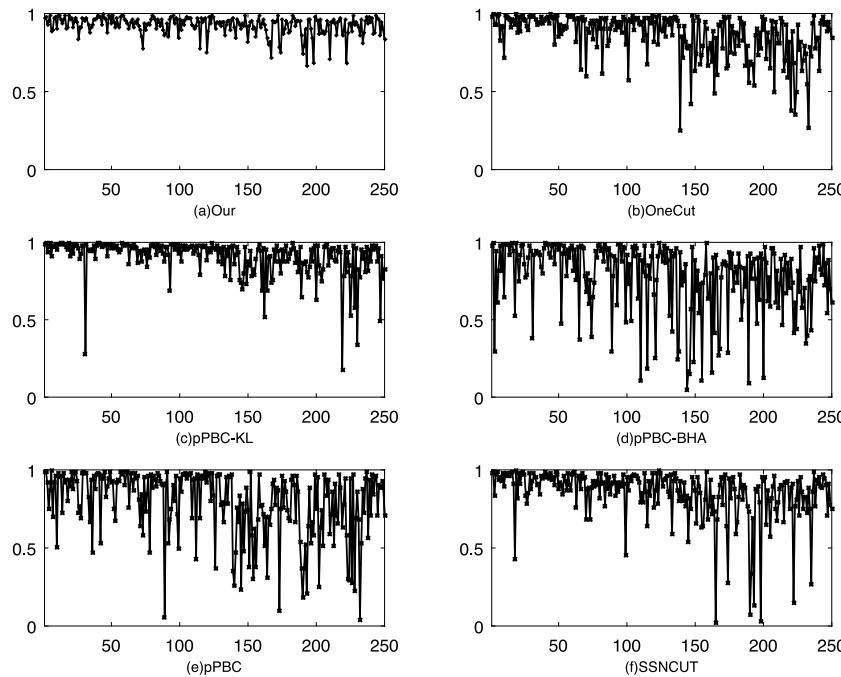


Fig. 8. Performance comparison of six typical image segmentation methods (evaluation index: Jac). The horizontal axis indicates the image number, and the ordinate indicates the Jaccard index. (a) Our method. (b) OneCut. (c) pPBC-KL. (d) pPBC-BHA. (e) pPBC. (f) SSNCUT.

Table 6
Average precision, recall, F_β and ME.

Method	Precision	Recall	F_β	ME
Our	0.9673	0.9533	0.9635	0.0121
OneCut	0.9558	0.9009	0.9361	0.0232
pPBC-KL	0.9673	0.9373	0.9557	0.0179
pPBC-BHA	0.9483	0.8183	0.8875	0.0473
pPBC	0.8354	0.9276	0.8464	0.0480
SSNCUT	0.9047	0.9340	0.9050	0.0269

seventh, and ninth lines, and a small number of isolated nodes appear in the 13th line. In the pPBC-KL method, lines 3, 5, 6, 11, 12, and 13 have some isolated nodes, and the head and legs of the sika deer in line 9 are not well segmented. In the OneCut method, there are some isolated nodes in most of the results. In the pPBC method, there are some isolated nodes in the 4th, 5th, 8th, 11th, 12th and 13th rows. There is an obvious segmentation error in line 5, line 7, line 8, and line 9. In the SSNCUT method, lines 4, 6, 9, 10 and 12 cannot completely segment the object. The experimental result shows that our method has better stability for simple and complex image segmentation.

4.5. Exchanging of seed point markers

To evaluate and compare several methods more objectively, we use the seed sets of our method, OneCut, and the SSNCUT method. Then, we exchange the marked points of the four methods in the experiment: (1) the OneCut method adopts the marked seed points in our method and the SSNCUT method for segmentation; (2) our method adopts the marked seed points in the OneCut and SSNCUT methods for segmentation; and (3) the SSNCUT method adopts the marked seed points in our method and the OneCut method for segmentation. The results are shown in Fig. 10 and Table 4.

From the segmentation results, the segmentation precision of OneCut and SSNCUT greatly decreases after the exchange of seed points.

Our method declines to some extent in precision, but it still maintains a good segmentation result. OneCut and SSNCUT have a good segmentation result when the foreground and the background are quite different. When the contrast between the object and the background is low or the texture is complex, the segmentation results are not good. In addition, there are more isolated point sets in the OneCut segmentation result, and more seed points need to be provided in the segmentation process.

4.6. Evaluation on the GrabCut dataset

We further compare five methods on the GrabCut dataset, where the seed points of the images are from Perret et al. (2015) (<https://perso.esiee.fr/~perretb/markerdb/>). These authors marked several seed sets of different qualities, and we selected the seeds “Marker_sk-sk.png” with fewer points as the seed points for segmentation. The experimental results are shown in Fig. 11 and Table 5.

The experimental results show that our method and RW have achieved better segmentation results, followed by pPBC, and the other methods are generally effective. Based on previous experiments, our method has a good segmentation effect and good stability.

4.7. Average precision, recall, F-measure, and ME

Table 6 shows the average precision, recall, F_β and ME values on the dataset of 250 images.

Table 7 shows the average precision, recall, F_β and ME values on the GrabCut dataset.

The experimental results show that our method produces better results than all the other methods.

4.8. Improvement in isolated points by the HPF algorithm

To verify the improvement in the pseudoflow algorithm compared to the graph cut method, this section conducts a comparative experiment on OneCut. The experimental method is as follows: first, the

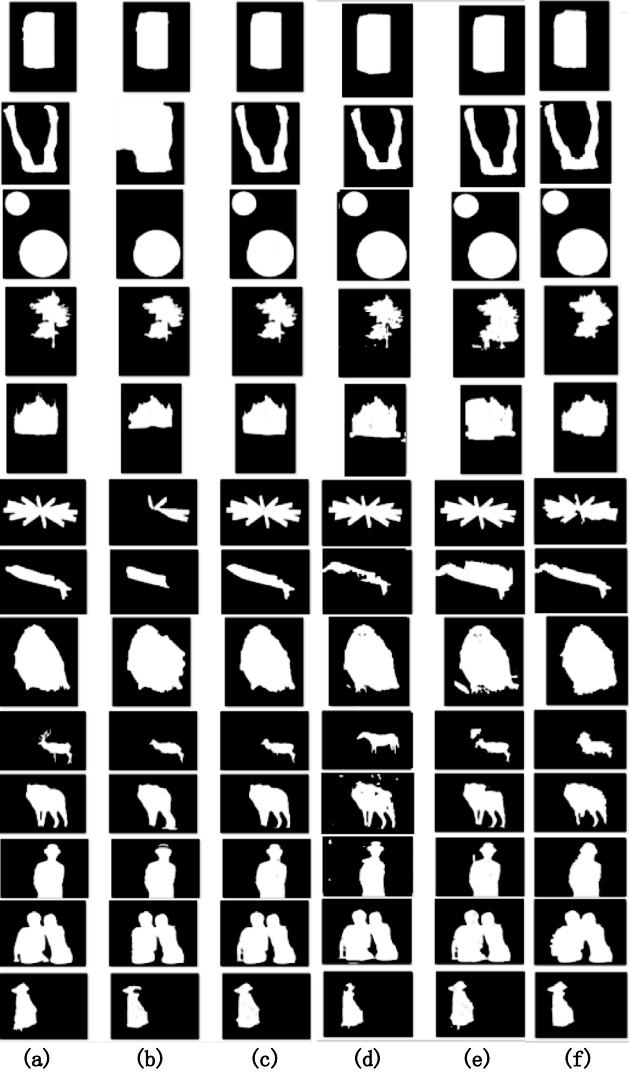


Fig. 9. Comparison of our method with six methods. The first column is the segmentation mask result of our proposed method, and columns 2–9 are the segmentation mask results of pPBC-BHA (b), pPBC-KL (c), OneCut (d), pPBC (e), and SSNCUT (f).

Table 7
Average precision, recall, F_β and ME.

Method	Precision	Recall	F_β	ME
Our	0.9752	0.9172	0.9605	0.0191
OneCut	0.9584	0.7129	0.7812	0.0576
RW	0.9569	0.9253	0.9475	0.0235
pPBC-KL	0.7120	0.8881	0.7408	0.0999
pPBC-BHA	0.7120	0.8881	0.7408	0.0999
pPBC	0.8629	0.9411	0.8730	0.0493
SSNCUT	0.6019	0.9817	0.6183	0.3794

model and code in OneCut are used to build the model and calculate the weight matrix. Then, graph cut and HPF are used for segmentation. The comparison of the segmentation results is shown in Fig. 12. The experimental results show that the OneCut algorithm using graph cut segmentation produces many isolated point sets, while that using the pseudoflow algorithm rarely has isolated point sets, and most of the images have a better segmentation effect than OneCut.

Table 8

OneCut segmentation results for different seed point markers. Jac is used as an evaluation indicator.

GrabCut dataset				
Method	Seeds	Avg.	Min.	Max.
OneCut(16^3 bins)	sk-sk	0.8896	0.5596	0.9856
OneCut(32^3 bins)	sk-sk	0.8896	0.5596	0.9856
OneCut(64^3 bins)	sk-sk	0.8896	0.5596	0.9857
OneCut(128^3 bins)	sk-sk	0.8896	0.5596	0.9859
OneCut(256^3 bins)	sk-sk	0.8897	0.5596	0.9867
Dataset of 250 images				
OneCut(16^3 bins)	OneCut	0.8373	0.3370	0.9960
OneCut(32^3 bins)	OneCut	0.8373	0.3370	0.9960
OneCut(64^3 bins)	OneCut	0.8375	0.3370	0.9960
OneCut(128^3 bins)	OneCut	0.8373	0.3370	0.9960
OneCut(256^3 bins)	OneCut	0.8373	0.3370	0.9960
OneCut(16^3 bins)	Our	0.8628	0.2157	0.9964
OneCut(32^3 bins)	Our	0.8628	0.2157	0.9964
OneCut(64^3 bins)	Our	0.8628	0.2157	0.9964
OneCut(128^3 bins)	Our	0.8635	0.2157	0.9964
OneCut(256^3 bins)	Our	0.8635	0.2157	0.9964
OneCut(16^3 bins)	SSNCUT	0.7902	0.0347	0.9964
OneCut(32^3 bins)	SSNCUT	0.7902	0.0347	0.9964
OneCut(64^3 bins)	SSNCUT	0.7902	0.0347	0.9964
OneCut(128^3 bins)	SSNCUT	0.7906	0.0347	0.9964
OneCut(256^3 bins)	SSNCUT	0.7906	0.0347	0.9964

4.9. Influence of noise on the segmentation method

We added Gaussian noise to some images and then performed segmentation. Fig. 13 shows the comparison of our method and OneCut. The experimental results show that OneCut produces many isolated nodes when the user inputs a box. OneCut requires more markup when the user input is scribbled. The effect of noise on our method is relatively small, and our method has good segmentation results.

In addition, the segmentation results in Figs. 4–6 show that our method successfully overcomes the defects in OneCut.

Fig. 14 shows part of the segmentation results of our algorithm, including the markers, segmentation results obtained on the original image and binary segmentation results.

4.10. Further comparison of the OneCut algorithm and our algorithm

In this section of the experiment, the OneCut seed point input method is consistent with that of our proposed method, including boxes and seed points. The input box is consistent with that of our proposed method. The energy function uses the first three terms in Eq. (7), and the code uses our implementation method. The results are shown in Table 8.

The experimental results show that when the OneCut method uses two types of user-marking manners at the same time, the segmentation accuracy is greatly improved on the GrabCut dataset, and the Jaccard index is slightly lower than that of our method. On a dataset of 250 images, when the input seed points are those of OneCut, the segmentation accuracy drops by 3% instead. When the input seed points are those of our method and SSNCUT, the segmentation accuracy is improved by nearly 10%, and the best value is lower than our accuracy by nearly 4%. Through modification of the OneCut energy function, there are very few isolated point sets in the segmentation result, which proves that the definition of the energy function has a greater impact on the generation of isolated point sets. In addition, for different seed point sets, the accuracy of the OneCut segmentation results is quite different, indicating that the OneCut algorithm is more sensitive to the position of the marked seed points.

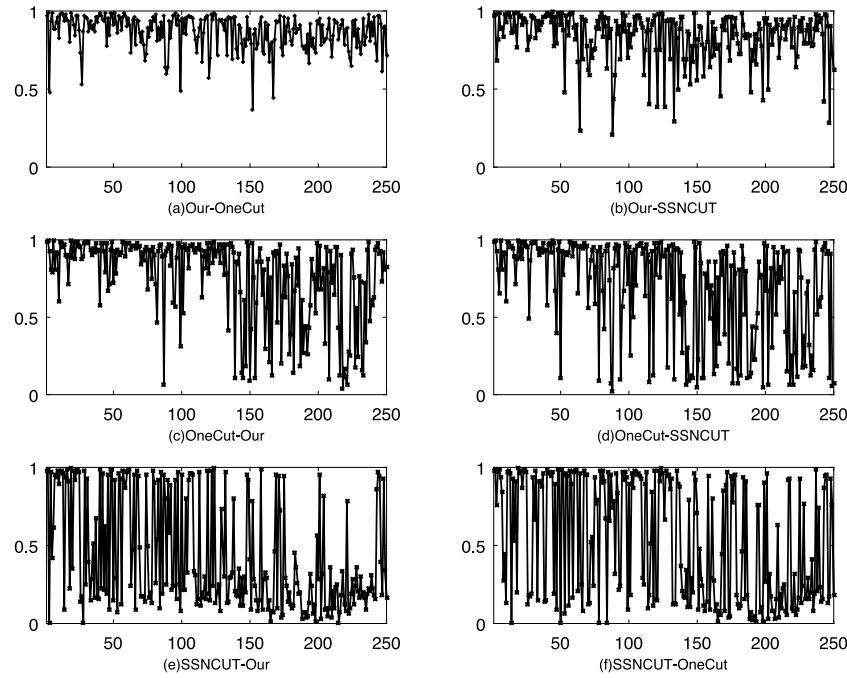


Fig. 10. Performance comparison of four image segmentation methods (evaluation index: Jac). The horizontal axis indicates the image number, and the ordinate indicates the Jaccard index. (a) Our method uses the OneCut seed set. (b) Our method uses the SSNCUT seed set. (c) OneCut uses our seed set. (d) OneCut uses the SSNCUT seed set. (e) SSNCUT uses our seed set. (f) SSNCUT uses the OneCut seed set.

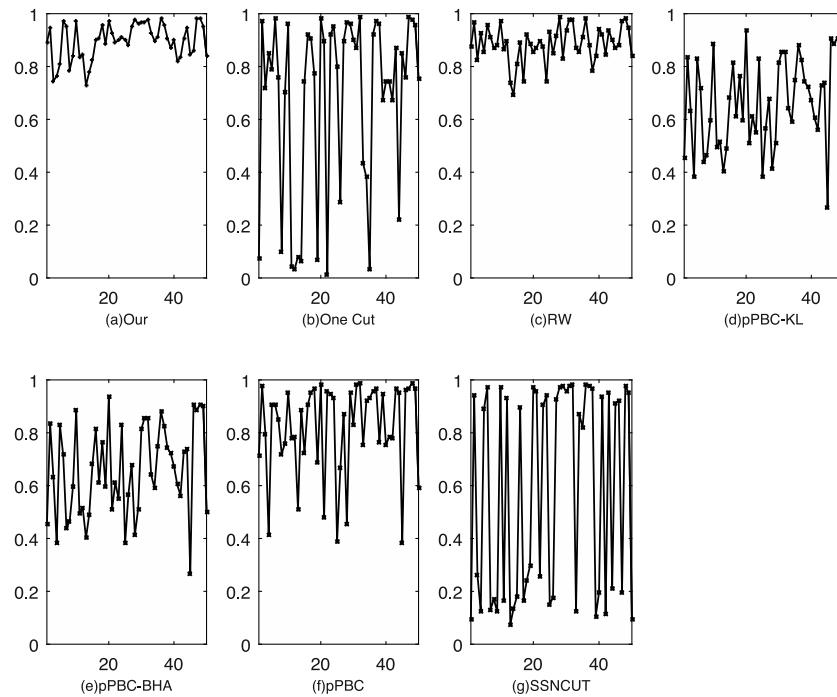


Fig. 11. Performance comparison of five typical image segmentation methods (evaluation index: Jac). The horizontal axis indicates the image number, and the ordinate indicates the Jaccard index. (a) Our method. (b) OneCut. (c) RW. (d) pPBC-KL. (e) pPBC-BHA. (f) pPBC. (g) SSNCUT.

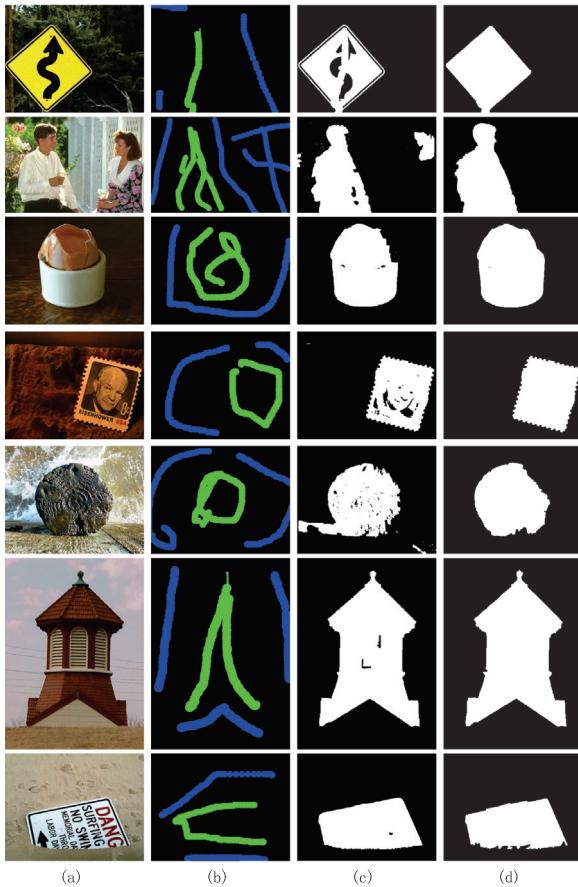


Fig. 12. Comparison of HPF algorithm and graph cut results: (a) original image, (b) user input foreground and background seeds, (c) segmentation result obtained by graph cut, and (d) segmentation result obtained by HPF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 9

Experimental comparison of the method proposed in this paper using HPF optimization and graph cut optimization. Jac is used as an evaluation indicator.

GrabCut dataset				
Method	Optimization	Avg.	Min.	Max.
Our(64 ³ bins)	Graph cut	0.8590	0.5675	0.9835
Our(64 ³ bins)	HPF	0.8971	0.7282	0.9835
Dataset of 250 images				
Our(64 ³ bins)	Graph cut	0.9012	0.3658	0.9924
Our(64 ³ bins)	HPF	0.9242	0.6617	0.9923

Below, we further verify the effect of the HPF method on improving segmentation. In the experiment, HPF and graph cut were used to optimize the method proposed in this paper, and the evaluation index used was the Jaccard index. The results are shown in Table 9.

The experimental results show that the segmentation accuracy of optimization using HPF is higher than that of optimization using graph cuts. On the GrabCut dataset, the segmentation accuracy is higher by nearly 4%. On the dataset of 250 images, the segmentation accuracy is higher by 2%. The experimental results prove that the HPF optimization improves the segmentation result accuracy.

Table 10

Segmentation speed comparison.

GrabCut dataset				
Method	Optimization	Avg.	Min.	Max.
OneCut	Graph cut	0.2407	0.0670	0.9140
Ours	HPF	2.1015	0.6360	4.6720
Ours	Graph cut	9.5264	1.5550	41.3860
RW	RW	0.5075	0.2310	0.7390
Dataset of 250 images				
OneCut	Graph cut	0.1869	0.0110	1.7220
Ours	HPF	1.0266	0.1750	3.2470
Ours	Graph cut	4.1603	0.3380	20.6180
RW	RW	0.4084	0.0400	0.7390

5. Segmentation speed comparison

In this section, we compare the segmentation times. First, the segmentation time of the proposed method (AMOE) is compared with those of OneCut and RW. Then, we compare the segmentation times of the AMOE method with different optimization methods. The experimental results are shown in 10.

According to the experimental results, the segmentation speed of OneCut is the fastest, the segmentation speed of RW is the second fastest, and the segmentation speed of AMOE is the slowest. The reason why AMOE has slower speed is the increase in the orientation energy in the energy function. By comparing the different optimization methods used in the AMOE method, the optimization method using the HPF algorithm is much faster than that using the graph cut algorithm, which also proves that the HPF segmentation is faster than the graph cut segmentation. Although the segmentation speed of AMOE has been reduced, the segmentation accuracy is better than that of other methods.

6. Conclusion

To overcome the shortcomings of OneCut, we introduced spatial distance and contour orientation energy constraints based on the energy term explicitly measuring the L_1 distance between the background and object appearance models in OneCut. We also modified the construction of energy graphs and replaced the graph cut algorithm with HPF to compute the maximum flow. The algorithm proposed in this paper has a good segmentation effect when there are similar colors in the object and the background, when the contrast between the object and the background is low, or when the texture is complex. In addition, the improved algorithm can segment any specified object in the image, and the user can further modify the segment by adding markers after obtaining the initial segmentation results. Thus, the algorithm is more suitable for practical applications.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the anonymous reviewers for their suggestions. The work was supported by the National Natural Science Foundation of China (No. 12071126); Hunan Provincial Science and Technology Project Foundation, China (No. 2018TP1018); and Scientific Research Fund of Hunan Provincial Education Department, China (19C1149). Shaojun Qu and Huang Tan contributed equally to this work.

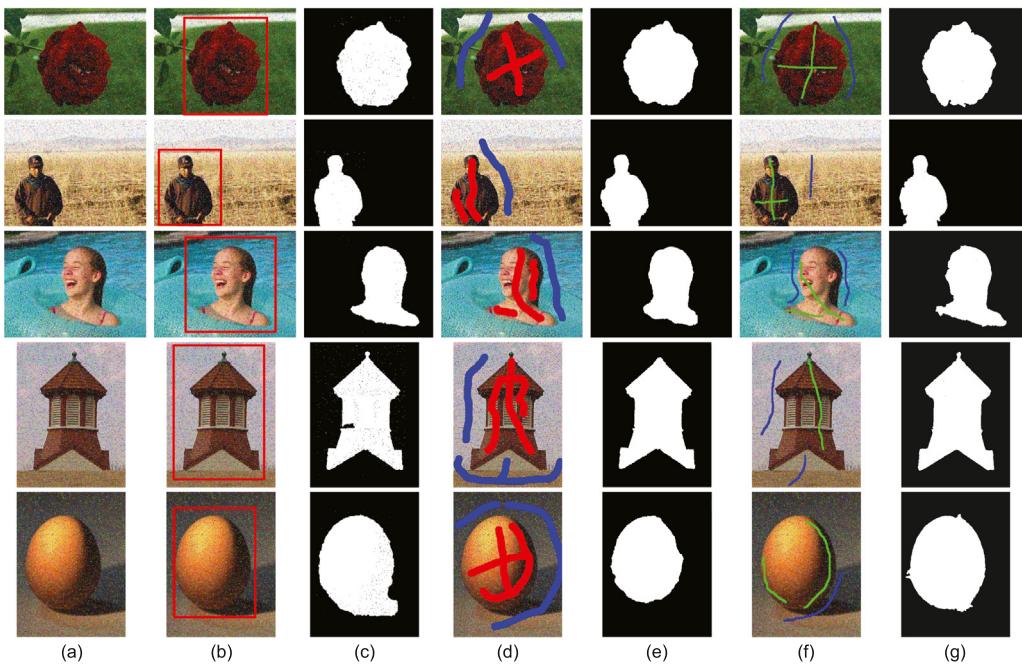


Fig. 13. Comparison of our method with OneCut: (a) Noise images, (b)–(c) OneCut with a box, (d)–(e) OneCut with seeds, and (f)–(g) our method. (b), (d) and (f) are the user input, and (c), (e) and (g) are the segmentation results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

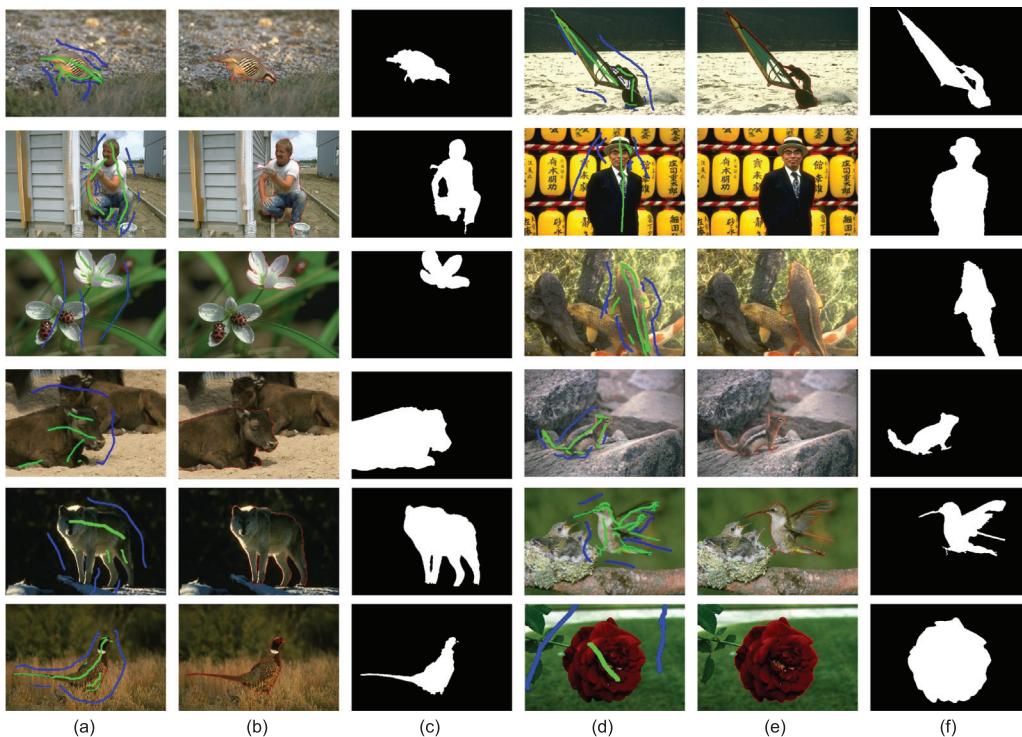


Fig. 14. Portion of the image segmentation results of our method. (a) and (d) are user input scribbled images, (b) and (e) are segmentation results, and (c) and (f) are mask images. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

- Bampis, C.G., Maragos, P., Bovik, A.C., 2017. Graph-driven diffusion and random walk schemes for image segmentation. *IEEE Trans. Image Process.* 26 (1), 35–50.
- Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (9), 1124–1137.
- Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (11), 1222–1239.
- Chandran, B.G., Hochbaum, D.S., 2009. A computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem. *Oper. Res.* 57 (2), 358–376.
- Chew, S.E., Cahill, N.D., 2015. Semi-supervised normalized cuts for image segmentation. In: 2015 IEEE International Conference on Computer Vision. pp. 1716–1723.
- Cox, I.J., Rao, S.B., Zhong, Y., 1996. “Ratio regions”: a technique for image segmentation. In: Proceedings of 13th International Conference on Pattern Recognition, Vol. 2, pp. 557–564.
- Ding, C.H.Q., He, X., Zha, H., Gu, M., Simon, H.D., 2001. A min-max cut algorithm for graph partitioning and data clustering. In: Proceedings 2001 IEEE International

- Conference on Data Mining, pp. 107–114.
- Dong, X., Shen, J., Shao, L., Van Gool, L., 2016. Sub-Markov random walk for image segmentation. *IEEE Trans. Image Process.* 25 (2), 516–527.
- Fishbain, B., Hochbaum, D.S., Mueller, S., 2016. A competitive study of the pseudoflow algorithm for the minimum $S-t$ cut problem in vision applications. *J. Real-Time Image Process.* 11 (3), 589–609.
- Grady, L., 2006. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (11), 1768–1783.
- Greig, D.M., Porteous, B.T., Seheult, A.H., 1989. Exact maximum a posteriori estimation for binary images. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 51, 271–279.
- He, K., Wang, D., Tong, M., Zhu, Z., 2020. An improved GrabCut on multiscale features. *Pattern Recognit.* 103, preprint.
- Hochbaum, D.S., 2008. The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Oper. Res.* 56 (4), 992–1009.
- Jain, A.K., Dubes, R.C., 1988. Algorithms for Clustering Data. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Lateef, F., Ruichek, Y., 2019. Survey on semantic segmentation using deep learning techniques. *Neurocomputing* 338, 321–348.
- Leung, T., Malik, J., 1998. Contour continuity in region based image segmentation. In: Burkhardt, H., Neumann, B. (Eds.), Computer Vision — ECCV'98. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 544–559.
- Liu, T., Sun, J., Zheng, N., Tang, X., Shum, H., 2007. Learning to detect a salient object. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8.
- Lovász, L., 1983. Submodular functions and convexity. In: Bachem, A., Korte, B., Grötschel, M. (Eds.), Mathematical Programming the State of the Art: Bonn 1982. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 235–257.
- Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vol. 2, pp. 416–423.
- McGuinness, K., O'Connor, N.E., 2011. Toward automated evaluation of interactive segmentation. *Comput. Vis. Image Underst.* 115 (6), 868–884.
- Peng, Z., Qu, S., Li, Q., 2019. Interactive image segmentation using geodesic appearance overlap graph cut. *Signal Process., Image Commun.* 78, 159–170.
- Peng, B., Zhang, L., Zhang, D., 2013. A survey of graph theoretical approaches to image segmentation. *Pattern Recognit.* 46 (3), 1020–1038.
- Perret, B., Copty, J., Ura, J.C.R., Guimarães, S.J.F., 2015. Evaluation of morphological hierarchies for supervised segmentation. In: Benediktsson, J.A., Chanussot, J., Najman, L., Talbot, H. (Eds.), Mathematical Morphology and its Applications to Signal and Image Processing. Springer International Publishing, Cham, pp. 39–50.
- Qu, S., Li, Q., Chen, M., 2019. Supervised image segmentation based on superpixel and improved normalised cuts. *IET Image Process.* 13 (12), 2204–2212.
- Ren, D., Jia, Z., Yang, J., Kasabov, N.K., 2017. A practical GrabCut color image segmentation based on Bayes classification and simple linear iterative clustering. *IEEE Access* (ISSN: 2169-3536) 5, 18480–18487.
- Rother, C., Kolmogorov, V., Blake, A., 2004. “GrabCut”: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23 (3), 309–314.
- Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8), 888–905.
- Shuai, B., Zuo, Z., Wang, B., Wang, G., 2018. Scene segmentation with dag-recurrent neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (6), 1480–1493.
- Tang, M., Ben Ayed, I., Boykov, Y., 2014. Pseudo-bound optimization for binary energies. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision — ECCV 2014. Springer International Publishing, Cham, pp. 691–707.
- Tang, M., Djelouah, A., Perazzi, F., Boykov, Y., Schroers, C., 2018. Normalized cut loss for weakly-supervised cnn segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1818–1827.
- Tang, M., Gorelick, L., Veksler, O., Boykov, Y., 2013. Grabcut in one cut. In: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1–8, 2013. IEEE Computer Society, pp. 1769–1776.
- Wu, Z., Leahy, R., 1993. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 15 (11), 1101–1113.