Journal of Electronic Imaging

JElectronicImaging.org

# Image segmentation via foreground and background semantic descriptors

Ding Yuan
Jingjing Qiang
Jihao Yin

SPIE. IS&T imaging.org

Ding Yuan, Jingjing Qiang, Jihao Yin, "Image segmentation via foreground and background semantic descriptors," *J. Electron. Imaging* **26**(5), 053004 (2017), doi: 10.1117/1.JEI.26.5.053004.

# Image segmentation via foreground and background semantic descriptors

**Ding Yuan, Jingjing Qiang, and Jihao Yin***
Beihang University, School of Astronautics, Beijing, China

**Abstract.** In the field of image processing, it has been a challenging task to obtain a complete foreground that is not uniform in color or texture. Unlike other methods, which segment the image by only using low-level features, we present a segmentation framework, in which high-level visual features, such as semantic information, are used. First, the initial semantic labels were obtained by using the nonparametric method. Then, a subset of the training images, with a similar foreground to the input image, was selected. Consequently, the semantic labels could be further refined according to the subset. Finally, the input image was segmented by integrating the object affinity and refined semantic labels. State-of-the-art performance was achieved in experiments with the challenging MSRC 21 dataset.  [DOI: 10.1117/1.JEI.26.5.053004]

Keywords: image segmentation; semantic label; nonparametric approach.

Paper 170292 received Apr. 18, 2017; accepted for publication Aug. 9, 2017; published online Sep. 9, 2017.

## 1 Introduction

Image segmentation is a fundamental problem in the field of computer vision. So far, abundant research has been published on this topic;[1–5] however, segmenting the complete foreground objects, which are not uniform in color or texture, remains a challenging task. In addition to the local low-level image features, such as color, texture, and spatial position, an increasing amount of studies focus on segmenting images using high-level visual information.

Cosegmentation methods suggested by Refs. 6–11 employ foreground correspondence and jointly segmented objects, which have similar characteristics in a set of images. Rother et al.[7] utilized histogram matching and a modified Markov random filed (MRF) framework formed by the difference of foreground region histograms. Sun et al.[8] constructed an MRF framework, which reflected camera flash illumination changes in order to extract the foreground from the background. Kim et al.[9] proposed a hierarchical framework for dividing the large image set into multiple subsets in order to perform segmentation by cosegmenting each subset separately with interimage connections. Inspired by the characteristic of linear anisotropic heat diffusion, Kim et al.[10] suggested a cosegmentation model, in which the finite heat sources of temperature maximization corresponded to the maximized segmentation confidence. In Ref. 11, segmentation was modeled by an energy-minimization function, which combined local appearance and spatial consistency; however, in most existing studies on cosegmentation, only multiple images with common objects were handled, and different irregularly appearing objects were hardly dealt with.

In recent years, semantic segmentation aiming at assigning a semantic label to each pixel of a given image[12–18] has become a subject undergoing intense investigation in the field of computer vision. Especially, the techniques of deep neural networks have recently played an important role in the field of semantic segmentation. The segmentation accuracy has been greatly improved by applying the deep learning techniques,[19–22] on the condition that the huge dataset is collected to train the network.

Semantic information, such as high-level visual information, can provide an important cue for the segmentation of a complete foreground from the image. In this study, inspired by semantic segmentation methods, we propose a segmentation mechanism for achieving a complete and accurate foreground boundary. Inspired by nonparametric methods, the initial semantic labels were obtained by maximizing the normalized label likelihood score.[23,24] Then, the foreground and background semantic descriptors were defined according to the initial semantic labels. With the aid of the two semantic descriptors, a subset of training images with similar foreground to the input image, was obtained. Subsequently, the semantic labels were further refined via object affinity and a semantic codebook. Finally, image segmentation was achieved by means of semantic labeling. For postprocessing, we adopted the Grab-Cut method[25] and used it to merge separate regions.

The remainder of this paper is organized as follows: Sec. 2 describes initial semantic labeling using the nonparametric method; Sec. 3 describes our image segmentation scheme via foreground and background semantic descriptors; and the experimental results are presented in Sec. 4.

## 2 Initial Semantic Labels Acquisition

Inspired by the nonparametric method,[23,24,26] the initial semantic labels of the input image can be acquired as follows: in the beginning, an image subset $\mathcal{D}$ is obtained from the training set by applying the global GIST feature descriptor, such that the image subset $\mathcal{D}$ contains the most scenes similar to the input image.

The GIST descriptor can summarize the gradient information for local regions of an image, which provides a rough description of the scene. A GIST descriptor of the scene refers to the meaningful information that an observer can

*Address all correspondence to: Jihao Yin, E-mail: yjh@buaa.edu.cn


**Fig. 1** Initial semantic labels throughout the image. (a) Superpixels and (b) coarse initial labels.

identify from a glimpse at the scene.[27] The GIST can be represented at both perceptual and conceptual levels because it includes all levels of visual information. It can be constructed by two-dimensional Gabor wavelets.[28] The Gabor wavelets of specific direction and scale can be considered as a local bandpass filter with respect to the corresponding direction and scale, whose response is exactly corresponding to the edges of specific directions in the image. At the beginning, the image is divided into patches. For each patch $P_i$ of size $r' \times c'$, the cascading of its convolution in each channel is defined as

$$G_i^P = \text{cat}[F(x,y) * G_{m,n}(x,y)], \quad (x,y) \in P_i, \tag{1}$$

where $\text{cat}(\cdot)$ represents the cascade operation, and $G_{m,n}(x,y)$ denotes the two-dimensional Gabor wavelet.

The average convolution of specific direction and scale for patch $P_i$ is $\bar{G}_i^P = \frac{1}{r' \times c'} \sum_{(x,y) \in P_i} G_i^P(x,y)$, then the GIST descriptor can be expressed as

$$G^G = \{\bar{G}_1^P, \bar{G}_2^P, \ldots, \bar{G}_{N_g}^P\}, \tag{2}$$

where $N_g$ represents the number of patches in the image.

By detecting and combining the edge information among local patches, the GIST descriptor can describe the overall distribution of gradient information within the image.

In this study, the initial semantic labels were assigned to each superpixel, instead of individual pixels, due to the spatial supports among pixels. Specifically, the superpixels, within both the input image and the images in $\mathcal{D}$, were obtained using the simple linear iterative clustering (SLIC) method.[29] Then, we adopted three features, denoted as $f^k$ ($k = 1, 2, 3$): the scale-invariant feature transform (SIFT) descriptor,[30] color mean in Lab color space, and central location of the superpixel, in order to describe each superpixel. $I_s = \{s_1, s_2, \ldots, s_N\}$ denotes the set of superpixels of the input image, and $D_s = \{n_1, n_2, \ldots, n_M\}$ denotes all the superpixels achieved from set $\mathcal{D}$. For each superpixel $s_i \in \{s_1, s_2, \ldots, s_N\}$, its neighborhood $\mathcal{N}_i^k$ was defined as a set of superpixels $\mathcal{N}_i^k \in D_s$, which had the nearest Euclidean distance to $s_i$ in terms of the $k$'th feature $f_i^k$. In this work, $\mathcal{N}_i^k$ included its closest 15 superpixels.

Next, each superpixel $s_i$ was assigned a semantic label $l \in L$, where $L$ represented the set of semantic classes. The probability distribution of semantic labels was defined as the normalized label likelihood score. In this work, the normalized label likelihood score $P(f_i^k|l)$, of each superpixel $s_i$, was expressed by nonparametric density estimates

$$P(f_i^k|l) = \frac{[n(l,\mathcal{N}_i^k) + \epsilon]/[n(l,\mathcal{D})]}{[n(\bar{l},\mathcal{N}_i^k) + \epsilon]/[n(\bar{l},\mathcal{D})]}, \tag{3}$$

where $\bar{l}$ was the complementary set of $l$. $n(l,\mathcal{N}_i^k)$ [or $n(l,\mathcal{D})$] and indicated the number of superpixels labeled $l$ in the set $\mathcal{N}_i^k$ (or $\mathcal{D}$), whereas $n(\bar{l},\mathcal{N}_i^k)$ [or $n(\bar{l},\mathcal{D})$] represented the number of superpixels that were not labeled $l$. The function of $\epsilon$ was to prevent zero likelihoods and smoothen the counts.

Then, the initial semantic label for each superpixel was achieved by maximizing the normalized label likelihood score

$$\max_l \sum_k \frac{1}{Z} P(f_i^k|l), \tag{4}$$

where $Z$ was a normalization factor under all sematic classes. However, the result of semantic labels was coarse and some superpixels were labeled incorrectly, as shown in Fig. 1.

## 3 Image Segmentation via Semantic Descriptors

With regard to the input image, it was intuitively known that the segmentation would be guided effectively if there existed a subset of training images, which would have a foreground similar to the input image. Therefore, the subset of training data, named semantic retrieval set, had to be determined by utilizing the initial semantic labels prior to complete segmentation.

### 3.1 *Semantic Retrieval Set Determination via Foreground and Background Semantic Descriptor*

In the beginning, the image could be divided into two segments according to its Lab color features by using the $k$-means method.[31] Considering that peripheral regions often appear as background in images, we assigned the peripheral segment, mentioned above, a background label "0"; a foreground label "1" was assigned to the other segment. Then, the foreground semantic descriptor $f_{\text{fs}}$ and the background semantic descriptor $f_{\text{bs}}$ were defined in order to obtain the semantic retrieval set $\Psi$, such that the images in the set would have the most similar foreground objects to the input image.

The semantic descriptors were defined in a spatial pyramid structure. The segment labeled "foreground," in the image, would be divided into equal grids with respect to different levels $r_s$ in the spatial pyramid. At each level, the semantic histogram was calculated within each grid. For instance, four semantic histograms $h_{21}$, $h_{22}$, $h_{23}$, and $h_{24}$ were obtained in the second level of the spatial pyramid, corresponding to the four equal grids, as shown in Fig. 2.

Then, the foreground semantic descriptor $f_{\text{fs}}$ was defined as the concatenation of all the semantic histograms within each grid at each pyramid level

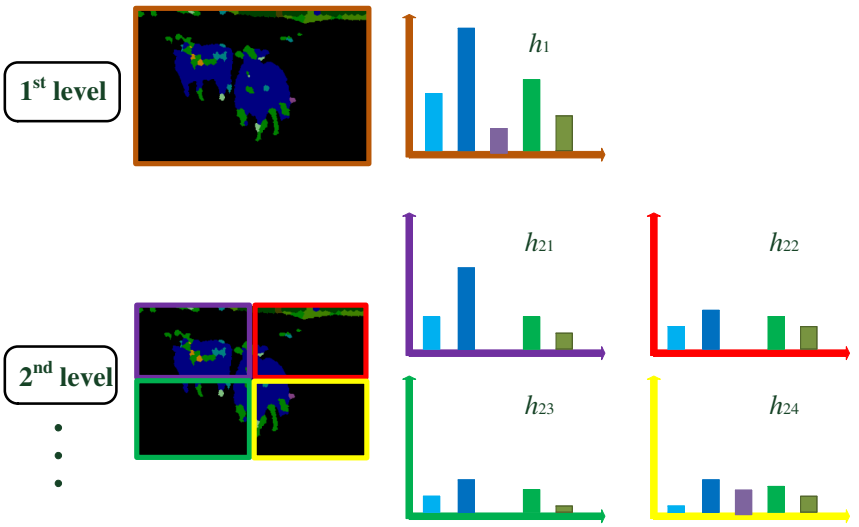


**Fig. 2** Definition of semantic descriptors. At each level, the semantic histograms were calculated within each grid.

$$f_{\mathrm{fs}} = [h_1, h_{21}, h_{22}, h_{23}, h_{24}, \ldots, h_{r_s, 4^{r_s-1}}]. \quad (5)$$

Similarly, we can also define the background semantic descriptor $f_{\mathrm{bs}}$. Experiments showed that the spatial pyramid layer $r_s = 3$ was a good compromise between capturing enough details and avoiding being sensitive to the noise.

In order to obtain the semantic retrieval set, we calculated the global GIST feature $g_f$, foreground semantic descriptor $f_{\mathrm{fs}}$, and background semantic descriptor $f_{\mathrm{bs}}$, throughout the input image and all the images in the training set. Then, the similarity of the input image and the training set was defined as the Euclidean distance between the features

$$d = d_g + d_{\mathrm{fs}} + \lambda d_{\mathrm{bs}}, \quad (6)$$

where $d_g$, $d_{\mathrm{fs}}$, and $d_{\mathrm{bs}}$ were the Euclidean distances, with respect to $g_f$, $f_{\mathrm{fs}}$, and $f_{\mathrm{bs}}$, between the input image and the training set. The purpose of $\lambda$ was to control the influence of the background semantic labels in the case where some training images had been wrongly selected due to their large background area. The procedure of obtaining the semantic retrieval set $\Psi$ was described in Algorithm 1.

By applying Algorithm 1, the training images were arranged according to the ascending order of $d$, which corresponded exactly to the similarity of the input image. Finally, the semantic retrieval set $\Psi$ was obtained by selecting the images corresponding to the smallest $d$. More images contained in set $\Psi$ would provide more clues for labeling the input image, but they would also decrease the similarity to the input image; therefore, in this study, a maximum of four training images was selected for the formation of the semantic retrieval set $\Psi$.

Figure 3 shows the semantic retrieval set $\Psi$ corresponding to the input "cow" image. Set $\Psi$ also had "cows" appearing in the foreground, which meant that the coarse initial semantic labels were able to provide an effective cue on what semantic categories the foreground belonged to.

**Algorithm 1.** Semantic retrieval set $\Psi$ selecting scheme.

1. Initialize $\lambda^{(0)} = 0$, and compute $d^{(0)} = d_g + d_{\mathrm{fs}} + \lambda^{(0)} d_{\mathrm{bs}}$;
2. Search for a subset $\Psi$ in an ascending order of distance $d^{(0)}$;
3. Set $t = 0$;
4. **while** none of background label of subset $\Psi$ is the same as the input image's background && $\lambda^{(t)} \neq 0.1$ **do**
5. $\lambda^{(t+1)} = \lambda^{(t)} + 0.01$
6. **if** $\lambda^{(t+1)} \neq 0.1$ **then**
7. $d^{(t+1)} = d_g + d_{\mathrm{fs}} + \lambda^{(t+1)} d_{\mathrm{bs}}$;
8. Search a new subset $\Psi$ in an ascending order of distance $d^{(t+1)}$;
9. $t = t + 1$
10. **end if**
11. **end while**
12. **return** $\lambda^{(t)}$, compute $d^{(t)} = d_g + d_{\mathrm{fs}} + \lambda^{(t)} d_{\mathrm{bs}}$, and obtain final subset $\Psi$.

### 3.2 *Semantic Labels Assignment via Object Affinity*

Once set $\Psi$ was obtained, the semantic labels would be reassigned to each superpixel of the input image via object affinity.

Suppose $s_i$ is a superpixel of the input image, and $s_{m_j}$ is a superpixel in the $m$'th image of $\Psi$; that is, $m \in \Psi$. We denoted the distance between $s_i$ and $s_{m_j}$ with respect to the $k$'th feature $f_i^k$ as $\Delta d_{im_j}^k$. Then, the distance measure between $s_i$ and $s_{m_j}$ was defined as

$$\Delta d_{im_j} = \sum_k \omega^k \Delta d_{im_j}^k, \quad (7)$$

where $\omega = [\omega^1, \omega^2, \omega^3]$ was the weight of different features; in this study, $k = 1, 2, 3$ as mentioned in Sec. 2.

(a) (b)

**Fig. 3** (b) The semantic retrieval set of the (a) input image.

Subsequently, within image $m$, the nearest neighborhood $\mathcal{N}_i^m$ of $s_i$ in set $\Psi$ was obtained via $\Delta d_{im_j}$.

Obviously, the semantic labels of $\mathcal{N}_i^m$ should have provided an important cue for assigning semantic labels to $s_i$, due to their high feature similarity. Moreover, the labels of the superpixels neighboring to $s_i$ in the input image should have obeyed the smoothness constraint, which reflected the distribution of semantic labels in natural images. The above idea can be expressed as the concept of object affinity.

Thus, the semantic label likelihood of $s_i$, determined by the labels of $\mathcal{N}_i^m$, was described as a Gaussian function

$$Gs(s_i) = \frac{1}{K_l}\sum_l \exp\left(-\frac{|\Delta d_{im_j}|}{2\beta_1^2}\right)\delta_{\mathcal{N}_i^m}(m_j), \tag{8}$$

where $K_l$ was the number of superpixels sharing the same semantic class $l$ within the neighborhood $\mathcal{N}_i^m$; $\beta_1$ was a damping parameter. The indicator function $\delta_{\mathcal{N}_i^m}(m_j)$ was defined as

$$\delta_{\mathcal{N}_i^m}(m_j) = \begin{cases} 1, & \text{if } s_{m_j} \epsilon \mathcal{N}_i^m \\ 0, & \text{otherwise} \end{cases}.$$

Considering that the distribution of semantic labels tended to be smooth throughout natural images, we adopted the agglomerative clustering method[10] in order to cluster the superpixels with respect to the Lab color feature. The semantic label propagation of the neighboring superpixels was achieved via object affinity

$$S_{\text{object}}(s_i) = \frac{4}{5}\cdot\frac{1}{|\mathcal{C}(s_i)|}\cdot\sum_{p\epsilon\mathcal{C}(s_i)} Gs(p) + \frac{1}{5}\cdot Gs(s_i), \tag{9}$$

where $\mathcal{C}(s_i)$ was the cluster where $s_i$ belonged.

### 3.3 *Initial Semantic Labels Refinement via Semantic Codebook*

Although the initial semantic labels were coarse, they still offered a strong cue about the distribution of semantic labels. Now that the semantic retrieval set $\Psi$ could also provide the probability of labels for each superpixel, we compared the similarity between initial semantic labels and semantic labels generated from the semantic retrieval set. Generally, the higher the similarity of the two semantic labels, the higher was the reliability of the semantic labels.

Hence, the initial semantic labels were refined according to a semantic codebook, which was constructed for measuring the similarity between initial semantic labels and the semantic labels in the semantic retrieval set $\Psi$. The semantic codebook of set $\Psi$ was set as $\mathcal{B} = \{\mathcal{B}_l^k | k = 1, 2, 3. l \in L_\Psi\}$, where $\mathcal{B}_l^k$ was defined as the feature descriptor of all the superpixels labeled $l$ in the $k$'th ($k = 1, 2, 3$) feature channel (mentioned in Sec. 2) for a specific semantic class $l$; $L_\Psi$ represented the set of semantic classes in set $\Psi$. Moreover, for any particular superpixel $s_j$ labeled $l$, its feature $f_{lj}^k$ ($k = 1, 2, 3$) formed a codeword in the codebook.

For any superpixel $s_i$ assigned a label $l_m$ initially, if its initial label $l_m$ was included in $L_\Psi$, such that $l_m \in L_\Psi$, the similarity of label $l_m$ was calculated only with respect to $\mathcal{B}_{lm} = \{\mathcal{B}_{lm}^k | k = 1, 2, 3\}$ in the semantic codebook. However, if the initial label $l_m \notin L_\Psi$, the similarity was determined by examining all the codewords in $\mathcal{B}$. Specifically, the similarity for superpixel $s_i$, which was initially labeled $l_m$, was defined as

$$\Delta H_i = \begin{cases} \sum_k \omega^k \Delta H_i^k & \text{if } l \in L_\Psi \\ \min_l \sum_k \omega^k \Delta H_{il}^k & \text{if } l \notin L_\Psi \end{cases}, \tag{10}$$

where $\Delta H_i^k$ recorded the smallest distance between feature $f_i^k$ of the superpxiel $s_i$ and the codeword in $\mathcal{B}_{lm}^k$. $\Delta H_{il}^k$ was the smallest distance between feature $f_i^k$ and a particular codeword in the semantic codebook $\mathcal{B}$.

Finally, for the initial semantic label of the superpxiel $s_i$, the semantic probability was refined as

$$S_{\text{refine}}(s_i) = \exp\left(-\frac{1}{|R_i|}\cdot\frac{|\Delta H_i|}{2\beta_2^2}\right), \tag{11}$$

where $|R_i|$ was the size of the semantic region of superpixel $s_i$ in the initial semantic labels; $\beta_2$ was a damping parameter. In this work, we set the parameters as $\omega = [\omega^1, \omega^2, \omega^3] = [0.3, 1, 0.3]$, $\beta_1 = 1$ and $\beta_2 = 0.035$.

### 3.4 *From Semantic Labeling to Segmentation*

In this section, we describe image segmentation by maximizing the linear combination of the object affinity and the refined probability of the initial semantic labels

$$V_{\text{seg}} = \max_l [S_{\text{object}}(s_i) + S_{\text{refine}}(s_i)]. \tag{12}$$

The segmentation results with the semantic labels of sample images, shown in Fig. 4(b), also provided a bounding box indicating the possible foreground area. Finally, the GrabCut method[25] was adopted as a postprocessing procedure to portray the boundary of the foreground precisely. The


**Fig. 4** Segmentation results by using Grab-Cut method[25] with respect to different artificial rectangular regions marked with green rectangles. (a) Segmentation results with respect to the precisely placed rectangle and (b) segmentation results with respect to the imperfect rectangle which do not encircle the foreground completely.

Grab-Cut method[25] is a segmentation technique that uses graph cuts to perform segmentation. Before it is performed, a manually rectangular region of interest should be placed to indicate the location of the foreground in the image. The more precisely the rectangle could exactly encircle the object of interest, the more accurate the segmentation result is. If the rectangular region of interest is not perfectly placed, the good segmentation result cannot be obtained, as shown in Fig. 4.

In this work, the semantic information can provide a bounding box indicating the possible foreground area. The Grab-Cut method is performed to further merge the neighboring regions assigned to the same semantic label in the bounding box, and at the same time, to precisely portray the boundary of the foreground via color features in the Lab space. Moreover, after performing the Grab-Cut method, for the areas where the new labels are not consistent with the segmentation result by using Eq. (12), the higher $V_{\text{seg}}$ will enforce these areas to remain their previous labels. In the experiments, we set the threshold as 1.4. Figure 5 shows the experimental results by using our method with Grab-Cut as the postprocessing procedure, and the results by using the Grab-Cut method with respect to a precisely placed artificial rectangular region of interest. Compared to Fig. 5(c), it shows that the our result with Grab-Cut as postprocessing procedure is more crisp and the boundary of the foreground is more precisely portrayed as shown in Fig. 5(d). Also in Fig. 5(d), the green leaves among the red flowers in the middle of the image is correctly labeled as the background, whereas it is wrongly labeled as the foreground by only applying the Grab-Cut method, as shown in Fig. 5(e).

More samples of postprocessing results are shown in Fig. 4(c). The foreground boundaries were eventually determined (Fig. 6).

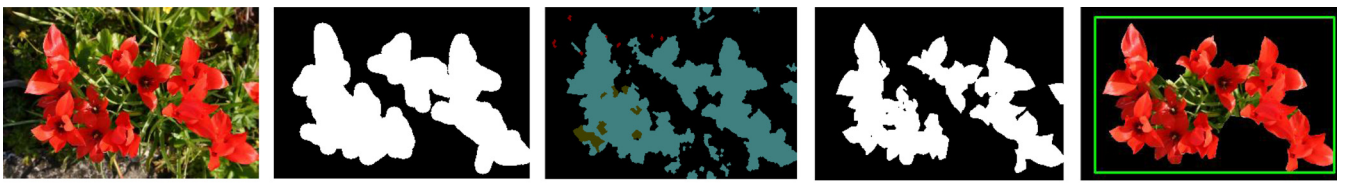


**Fig. 5** Segmentation results by using our method and by directly using the Grab-Cut method. (a) Input image, (b) ground truth, (c) segmentation results with semantic labels, (d) our segmentation result with Grab-Cut as a postprocessing procedure, and (e) result by using Grab-Cut with respect to a manually rectangular region of interest marked in green.

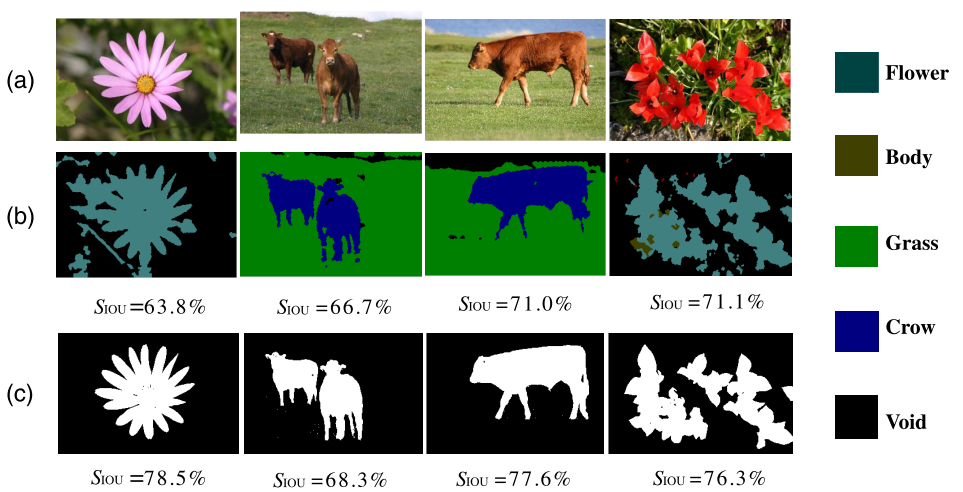


**Fig. 6** Segmentation results. (a) Input images, (b) segmentation results with semantic labels, and (c) postprocessing using Grab-Cut.

## 4 Experimental Results

To verify the effectiveness of the proposed method, we conducted experiments on the MSRC 21 dataset, which contained 21 different classes with 276 training images and 256 testing images. In the experiments using the MSRC 21 dataset, the image subset $\mathcal{D}$ was allowed to include a maximum of 25 training images, such that enough scenes similar to the input image could be selected.

The segmentation performance was validated via the intersection-over-union score mentioned in Refs. 10 and 12

$$S_{\mathrm{IOU}} = \max_l \frac{1}{|I|} \sum_{i \in I} \frac{\mathrm{GT}_i \cap R_i^l}{\mathrm{GT}_i \cup R_i^l}, \tag{13}$$

where $\mathrm{GT}_i$ is the ground truth and $R_i^l$ represents the region associated with $l$'th class in image $i$.

In the experiments, we tested all of the 14 image classes in the MSRC 21 dataset. The intersection-over-union score was adopted in order to evaluate the precision of our algorithm. Moreover, we compared our algorithm to other segmentation algorithms;[2,6,10–12,18,22] the results are listed in Table 1. The higher intersection-over-union score corresponded to higher segmentation precision. In addition, we also listed the precision of our semantic labeling named as "semantic label," and the segmentation results by using the initial labels with the Grab-Cut as postprocessing is named as "initial + grab cut" in Table 1. "Ours" represents the results of our final segmentation with the Grab-Cut as postprocessing. The subscript represents the rank of segmentation accuracy by using different methods.

We also compared our results with the technique of deep neural network.[22] In the experiment, we directly evaluated the released pretrained model trained with PASCAL-context dataset on the MSRC dataset and computed the segmentation accuracy of nine overlapping classes between the two datasets. The quantitative results have been listed in Table 1. Admittedly, the average accuracy of our results is lower than that of Ref. 22, which involved a large amount of training samples. However, our results have achieved the best average accuracy among the hand-designed feature-based methods. In addition, the segmentation accuracies of several classes, such as "cow," "dog," and "sheep" are comparable to that of Ref. 22. We also achieved a better result on class "chair" compared to Ref. 22. Details on the experimental results are listed in Table 1. Figure 7 shows segmentation results by using our proposed algorithm. We also compared our results with Ref. 22 visually, as shown in Figs. 7(c) and 7(f). For the results by using,[22] only the overlapping classes between the MSRC and PASCAL-Context datasets are shown. For example, segmentation results by using[22] are not shown from the 1st row to the 4th row in Fig. 7, as the PASCAL-Context dataset does not include the classes of "face", "flower", "sign" and "house", which is also illustrated in Table 1.

In the experiments, there are still some images which are very challenging. Our mechanism of semantic labeling and

**Table 1** Segmentation results evaluation on intersection-over-union score.

| Class | Ours | Initial + Grab-Cut | Semantic label | Ref. 12 | Ref. 11 | Ref. 10 | Ref. 6 | Ref. 2 | Ref. 18 | Ref. 22 |
|---|---|---|---|---|---|---|---|---|---|---|
| Bike | $40.3_5$ | $42.1_4$ | $36.3_7$ | $39.9_6$ | $43.3_2$ | $29.9_8$ | $42.8_3$ | $13.7_{10}$ | $27.0_9$ | $59.5_1$ |
| Bird | $62.6_2$ | $50.9_3$ | $33.5_7$ | $48.3_4$ | $47.7_5$ | $29.9_8$ | - | $34.3_6$ | $20.2_9$ | $71.8_1$ |
| Car | $68.2_2$ | $52.7_4$ | $48.0_7$ | $52.3_6$ | $59.7_3$ | $37.1_8$ | $52.5_5$ | $20.1_{10}$ | $35.3_9$ | $90.7_1$ |
| Cat | $57.6_3$ | $58.6_2$ | $41.5_5$ | $52.3_4$ | $31.9_7$ | $24.4_8$ | $5.6_{10}$ | $33.5_6$ | $19.9_9$ | $85.1_1$ |
| Chair | $61.0_1$ | $53.1_3$ | $22.8_{10}$ | $54.3_2$ | $39.6_5$ | $28.7_7$ | $39.4_6$ | $24.1_9$ | $24.5_8$ | $49.0_4$ |
| Cow | $74.6_2$ | $63.9_3$ | $50.6_5$ | $43.2_7$ | $52.7_4$ | $33.5_8$ | $26.1_9$ | $44.8_6$ | $15.7_{10}$ | $78.5_1$ |
| Dog | $79.9_2$ | $60.6_3$ | $30.6_8$ | $50.8_4$ | $41.8_6$ | $33.0_7$ | - | $43.6_5$ | $22.5_9$ | $84.6_1$ |
| Face | $55.2_3$ | $58.7_2$ | $42.2_6$ | $45.8_5$ | $70.0_1$ | $33.2_8$ | $40.8_7$ | $48.3_4$ | $28.1_9$ | - |
| Flower | $66.1_2$ | $58.6_3$ | $43.7_5$ | $84.9_1$ | $51.9_4$ | $40.2_6$ | - | $26.8_8$ | $39.6_7$ | - |
| House | $63.0_2$ | $44.9_5$ | $38.9_7$ | $48.6_4$ | $51.0_3$ | $32.3_8$ | $66.4_1$ | $28.4_9$ | $44.4_6$ | - |
| Plane | $46.0_2$ | $40.9_4$ | $44.1_3$ | $35.9_5$ | $21.6_9$ | $25.1_7$ | $33.4_6$ | $25.0_8$ | $16.7_{10}$ | $70.8_1$ |
| Sheep | $78.3_2$ | $70.8_3$ | $56.0_7$ | $66.3_4$ | $66.3_4$ | $60.8_6$ | $45.7_8$ | $38.0_{10}$ | $45.1_9$ | $85.3_1$ |
| Sign | $84.8_1$ | $77.1_2$ | $46.1_6$ | $59.5_3$ | $58.9_4$ | $43.2_7$ | - | $42.4_8$ | $58.6_5$ | — |
| Tree | $74.1_1$ | $65.7_4$ | $66.8_3$ | $58.1_7$ | $67.0_2$ | $61.2_6$ | $55.9_8$ | $30.4_9$ | $64.6_5$ | — |
| Average | $65.1_2$ | $57.0_3$ | $42.9_6$ | $52.9_4$ | $50.2_5$ | $36.6_8$ | $40.9_7$ | $32.4_9$ | $31.6_{10}$ | $75.1_1$ |

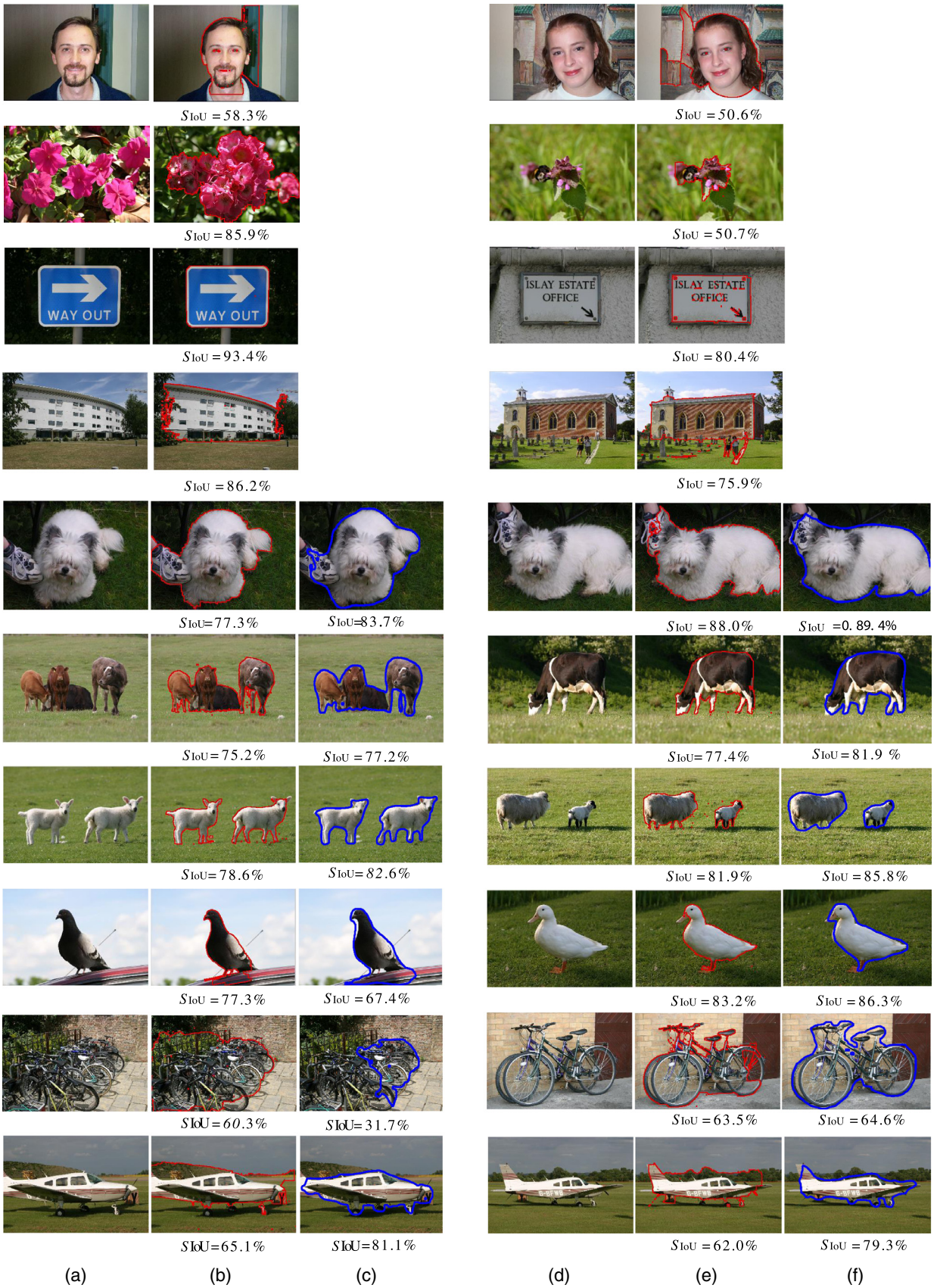


**Fig. 7** Sample segmentation results. (a) and (d) Input images, (b) and (e) segmentation results by using our proposed method, and (c) and (f) segmentation results by using the technique of deep neural network.[22]


**Fig. 8** Incorrect segmentation results. (a) and (c) Input images and (b) and (d) segmentation results. The segmentation failed because the color or texture distribution of the foreground is similar to the background.

foreground segmentation depends on the color and SIFT features. Consequently, our method would probably fail, if dealing with the images in which the color or texture distribution of the foreground is similar to the background, as shown in Fig. 8.

## 5 Conclusion and Future Work

In this paper, an image segmentation framework based on semantic information was proposed. Unlike traditional methods based on low-level features, we adopted semantic information in order to distinguish the foreground from the background. In our study, the initial semantic labels were obtained using the nonparametric method. By searching for similar images in the training data, the input image was segmented via the combination of object affinity and semantic labels. Experimental testing using the MSRC 21 dataset demonstrated that our method performed well. In future work, segmentation of video data by means of semantic information will be investigated.

### *Acknowledgments*

This work was supported by the National Natural Science Foundation of China (No. 61005031).

### *References*

1. D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 603–619 (2002).
2. P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision* **59**(2), 167–181 (2004).
3. Z. Kato and T.-C. Pong, "A Markov random field image segmentation model for color textured images," *Image Vision Comput.* **24**(10), 1103–1114 (2006).
4. S. Ramalingam, P. Kohli, and K. Alahari, "Exact inference in mufti-label CRFs with higher order cliques," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8 (2008).
5. Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(9), 1124–1137 (2004).
6. L. Mukherjee, V. Singh, and J. Peng, "Scale invariant cosegmentation for image groups," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2011).
7. C. Rother et al., "Cosegmentation of image pairs by histogram matching incorporating a global constraint into MRFs," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 993–1000 (2006).
8. J. Sun et al., "Flash cut: foreground extraction with flash and no-flash image pairs," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8 (2007).
9. E. Kim, H. Li, and X. Huang, "A hierarchical image clustering cosegmentation framework," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2012).
10. G. Kim et al., "Distributed cosegmentation via submodular optimization on anisotropic diffusion," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 169–176 (2011).
11. A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 542–549 (2012).
12. Y. Liu et al., "Weakly-supervised dual clustering for image semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2075–2082 (2013).
13. L. Tao, F. Porikli, and R. Vidal, "Sparse dictionaries for semantic segmentation," in *European Conf. on Computer Vision*, pp. 549–564, Springer International Publishing (2014).
14. J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: joint object detection, scene classification and semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 702–709 (2012).
15. C. Göring et al., "Semantic segmentation using GrabCut," in *Proc. of the Int. Conf. on Computer Vision Theory and Applications (VISAPP)* (2012).
16. P. Arbelaez et al., "Semantic segmentation using regions and parts," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3378–3385 (2012).
17. G. Csurka and F. Perronnin, "An efficient approach to semantic segmentation," *Int. J. Comput. Vision* **95**(2), 198–212 (2011).
18. J. Yang, B. Price, and S. Cohen, "Context driven scene parsing with attention to rare classes," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3294–3301 (2014).
19. H. Zhao et al., "Pyramid scene parsing network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2881–2890 (2017).
20. Y. Li et al., "Fully convolutional instance-aware semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2359–2367 (2017).
21. H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 1520–1528 (2015).
22. J. Long, E. Shelhamer, and T. Darrel, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 640–651 (2017).
23. G. Singh and J. Košecká, "Semantic context for nonparametric scene parsing and scene classification," in *Scene Understanding Workshop, CVPR* (2013).
24. G. Singh and J. Košecká, "Nonparametric scene parsing with adaptive feature relevance and semantic context," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3151–3157 (2013).
25. C. Rother et al., "'GrabCut': interactive foreground extraction using iterated graph cuts," *ACM Trans. Graphics* **23**(3), 309–314 (2004).
26. J. Tighe and S. Lazebnik, "SuperParsing: scalable nonparametric image parsing with superpixels," *Lect. Notes Comput. Sci.* **6315**, 352–365 (2010).
27. M. C. Potter, "Meaning in visual search," *Science* **187**, 965–966 (1975).
28. A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *Int. J. Comput. Vision* **42**(3), 145–175 (2001).
29. R. Achanta and A. Shaji, "SLIC superpixels compared to state-of-the art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2274–2282 (2012).
30. C. Liu et al., "SIFT flow: dense correspondence across scenes and its applications," *IEEE Trans Pattern Anal Mach Intell.* **33**(5), 978–994 (2011).
31. J. A. Hartigan and M. A. Wong, "A K-means clustering algorithm," *Appl. Stat.* **28**(1), 100–108 (1979).

**Ding Yuan** received her PhD in mechanical and automation engineering from the Chinese University of Hong Kong and has worked in the field of computer vision for over 15 years. She is now an associate professor at the Image Processing Center, School of Astronautics, Beihang University.

**Jingjing Qiang** is a postgraduate student at the Image Processing Center, School of Astronautics, Beihang University.

**Jihao Yin** received his PhD in computer science from Beijing Institute of Technology. He is now an associate professor at the Image Processing Center, School of Astronautics, Beihang University. His research interests include image processing and remote sensing.