IEICE TRANS. INF. & SYST., VOL.E97–D, NO.10 OCTOBER 2014

LETTER

Foreground Segmentation via Dynamic Programming

Bing LUO[†a)], **Chao HUANG**[†], **Lei MA**[†], **Wei LI**[†], *Nonmembers*, *and* **Qingbo WU**[†], *Student Member*

SUMMARY This paper proposes a novel method to segment the object of a specific class based on a rough detection window (such as Deformable Part Model (DPM) in this paper), which is robust to the positions of the bounding boxes. In our method, the DPM is first used to generate the root and part windows of the object. Then a set of object part candidates are generated by randomly sampling windows around the root window. Furthermore, an undirected graph (the minimum spanning tree) is constructed to describe the spatial relationships between the part windows. Finally, the object is segmented by grouping the part proposals on the undirected graph, which is formulated as an energy function minimization problem. A novel energy function consisting of the data term and the smoothness term is designed to characterize the combination of the part proposals, which is globally minimized by the dynamic programming on a tree. Our experimental results on challenging dataset demonstrate the effectiveness of the proposed method.

key words: *object segmentation, undirected graph, dynamic programming*

1. Introduction

Foreground segmentation is challenging since the foreground prior is usually lacked in the segmentation process. In the last decade, various object prior discovery methods have been used to achieve the foreground extraction, which can be roughly classified into two classes, i.e., unsupervised foreground segmentation [1]–[4] and supervised foreground segmentation [5]–[10]. While the former focuses on jointly discovering the foreground prior from the multiple images automatically, the later requires the pixel-level labels for the training set. Bounding box based segmentation method [6] is an important supervised method, which is usually used as the successor process of the object detection task. In the last decade, the bounding box based segmentation has been paid much attention, and several bounding box foreground segmentation methods have been proposed, such as GrabCut [6] and bounding box prior [11]. However, the successful object segmentation of these existing bounding box segmentation methods is usually based on the assumption that the bounding box has covered the object region well enough. In other words, when the bounding boxes do not cover the object region accurately enough, such as the usual outputs of the object detection methods that only partially contain the object regions, these methods will lead to unsuccessful object segmentation, which is usually happened in the practice

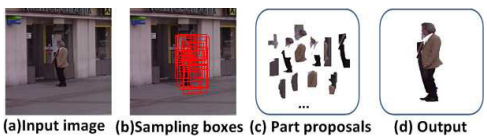

Fig. 1 The explanation of segmentation from the complex background.

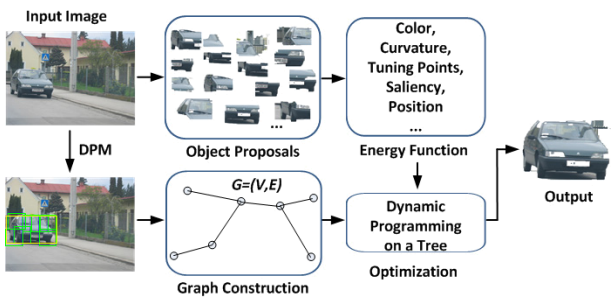

Fig. 2 The flowchart of the proposed method.

foreground extraction after the object detection process.

In this paper, we propose a novel method to accurately segment the foreground object from the rough bounding boxes obtained by the Deformable Part Model (DPM) object detection method [12]. Our method is motivated by the observation that the object or its local parts can be well segmented by moving and scaling the rectangle around the object via GrabCut. Thus, the accurate object extraction can be achieved by segmenting and combining these local segments using their similarities and the spatial relationships. The proposed method consists of four steps. In the first step, we detect the objects by DPM and obtain two types of windows, i.e., a root window and a set of part windows. To handle complex background, we randomly sample a set of boxes around the root window by varying the window's size and positions, and perform the GrabCut on the sampling boxes to obtain a large number of local part segment proposals, as shown in Fig. 1. In the second step, we construct a minimum spanning tree (MST) to describe the spatial relationships of the object local parts by the part windows. The foreground extraction is then formulated as selecting and combing a subset of segment proposals that best fit the local similarities and spatial relationships of the tree. Thirdly, based on the MST, an objective function consisting of unary term and smoothness term is defined to model the proposals selection. Finally, we use the dynamic programming method to minimize the objective function on the tree structure to achieve the foreground extraction. The

Manuscript received April 17, 2014.
Manuscript revised June 25, 2014.
[†]The authors are with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.
a) E-mail: Mathild1987@163.com
DOI: 10.1587/transinf.2014EDL8078

flowchart of the proposed method is illustrated in Fig. 2.

Our contributions include: (1) we propose a novel method to segment the foreground object from the rough bounding boxes obtained by DPM, which is robust to the positions of the bounding boxes. (2) Foreground segmentation is constructed by a minimum spanning tree structure which is minimized by dynamic programming techniques. (3) The segmentation quality evaluation is introduced to segment the foreground objects.

2. Proposed Method

2.1 Object Proposal Generation

Given an image I, we perform DPM [12] on the image and obtain a root detection window with a set of part detection windows. We denote them as $W = (w_0, w_1, w_2, \ldots, w_K)$, where w_0 is the root window, $w_i, 1 \leq i \leq K$ are the ith part window, and K is the number of the part windows. To handle the shape variations of the object and complex background, we randomly select N object windows around the center of the root window w_0 by varying its size and the central position. In the sampling, the centers of the sample windows are set to be uniformly distributed around the root window. Furthermore, the sizes of the windows (the width and the hight) are uniformly sampled, i.e., $W = \alpha W_r$, $H = \alpha H_r$ and $\alpha \sim \mathcal{U}(\frac{1}{4}, \frac{1}{2})$, where W_r and H_r are the width and hight of the root window, respectively. Based on the sampling windows, we use the GrabCut to obtain a set of segment proposals which is denoted as $U = \{u_i, 1 \leq i \leq N\}$.

2.2 Graph Construction

Based on the assumption that the local parts of the object are contained in the segment proposals U, the object segmentation can be formulated as properly selecting and combining the right segment proposals in U that best fit the shape structure of the object part regions.

To obtain the object shape structure, we construct graph $G = (V, E)$ to represent the spatial relationships between the object parts, where V and E are the node set and edge set respectively. In the graph generation, we first generate the nodes of the graph based on the given part windows, i.e., $V = \{v_1, \ldots, v_K\}$ and v_i represents w_i. Then, the edges $e = (v_i, v_j)$ between each node pair (v_i, v_j) are added to describe the relationship between the local window pair (w_i, w_j). Each edge $e = (v_i, v_j)$ is assigned a weight ω_{ij} to represent the relationship. We set $\omega_{ij} = \|z_i - z_j\|$ as the spatial distance between the windows (w_i, w_j), where z_i and z_j are center position of w_i and w_j, respectively. It is seen that the constructed graph is fully connected.

Based on G, we next search the MST Q on the graph to clearly describe the structure model of the object parts, as shown in Fig. 3. The advantage of the minimum spanning tree structure is that there are no cycles so that it is able to benefit and simplify the selection information propagation. Based on Q, the segmentation problem changes to search the object proposal x_i for each node v_i that not only x_i fulfill v_i, but also the proposal pairs (x_i, x_j) satisfies the relationships between (v_i, v_j). Here, we propose a new energy minimization model to obtain $X = \{x_1, \ldots, x_K\}$. We next detail the energy function generation and minimization.

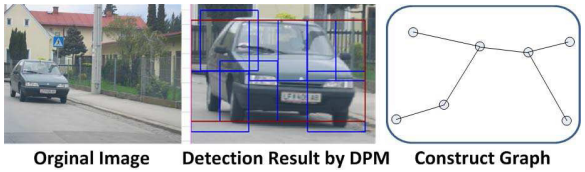

Fig. 3 The graph construction. The left: the original image. The center: the detection results by DPM. The right: the MST constructed by the spatial relationships of the part windows.

2.3 The Energy Function

We model the selection $X = \{x_1, \ldots, x_K\}$ as

$$E(x_1, \ldots, x_K) = \sum_{v_i \in V} D_i(x_i) + \sum_{(v_i, v_j) \in E} V_{i,j}(x_i, x_j) \tag{1}$$

where D and V are the unary term and pairwise term, respectively. K is the number of local parts in the tree.

2.3.1 The Unary Potential D

The unary potential is used to describe the fitness between the proposal u_{x_i} and the corresponding local part w_i. We consider four terms to define D, i.e.,

$$D_i(x_i) = E_{sal}(x_i) + E_{curv}(x_i) + E_{tp}(x_i) + E_{pos}(x_i) \tag{2}$$

The details of the terms are described as follows.

(1) The Saliency Term E_{sal}

As we know, the target objects are usually the salient object. The proposal covering the salient regions tends to be the local part of the object. Let A be the image saliency map. We calculate saliency value S_i of the proposal u_{x_i} by [3]:

$$S_i = \frac{\sum_{p \in u_{x_i}} A(p)}{m_i} \cdot \frac{m_i'}{M'} \tag{3}$$

where m_i is the number of non-zero pixels in u_{x_i}. M' is the number of the salient pixels in the image, and m_i' is the number of the salient pixels in the proposals u_{x_i}. We consider the pixel as the saliency pixel when its saliency value is larger than threshold $T_s = 0.3$. $\frac{m_i'}{M'}$ is introduced to avoid the small regions with large mean saliency value. Based on S_i, we define the saliency term as:

$$E_{sal}(x_i) = 1 - normalized(S_i) \tag{4}$$

where $normalized(\cdot)$ is the normalized function.

In our method, the saliency map A is efficiently calculated by overlapping the segmentation mask of the proposals U. Let $A(p)$ denote the image saliency value at pixel p, the saliency map is obtained by:

$$A(p) = \frac{1}{N_z}\sum_{l=1}^{N}\delta(u_l(p)) \tag{5}$$

where N_z is a normalized constant, N is the number of proposals.

(2) The Boundary Curvature Term E_{curv}

It is observed that good segments usually have smooth boundaries. On the contrary, a rough boundary corresponds to bad segments. Motivated by such observation, we use boundary smoothness to select the right potentials. The boundary smoothness is represented by the boundary curvature E_{curv}, which is defined as [13], [14]:

$$E_{curv}(x_i) = curv(u_{x_i}) = \sum_k \frac{\det(u''_{x_i}(p_k))}{(1+(u'_{x_i}(p_k))^2)^{\frac{3}{2}}} \tag{6}$$

where $u''_{x_i}(p_k)$ and $u'_{x_i}(p_k)$ are the Hessian matrix and the first-order derivation at pixel p_k of proposal u_{x_i}, respectively.

(3) The Boundary Turning Points Term E_{tp}

By observing the boundary of the segment proposal u_{x_i}, it is concluded that the boundary of a bad segment usually has many turning points. Motivated by [15], we introduce boundary turning points term depicted as:

$$E_{tp}(x_i) = \frac{\sum_k \delta((u_{x_i} * \phi)(p_k) < \eta_{x_i})}{|C|} \tag{7}$$

where ϕ is the low-pass filter, and $(u_{x_i} * \phi)(p_k)$ is the response of the low-pass filter on pixel p_k. η_{x_i} is the threshold, C is the set of boundary pixels p_k.

(4) The Position Term E_{pos}

We intend to select the proposals near the corresponding local part window. The position term E_{pos} is introduced for the near proposal selection, which is defined as the spatial distance between the centers of x_i and part window w_i, i.e.,

$$E_{pos}(x_i) = d(z_{u_{x_i}}, z_{w_i}) \tag{8}$$

where $z_{u_{x_i}}$ and z_{w_i} are the center position of u_{x_i} and w_i, respectively. $d(\cdot)$ is the normalized Euclidean distance.

2.3.2 The Smoothness Potential V

The smoothness potential V is used to punish the two proposals that do not satisfy the spatial relationships. We define this evaluation by two terms represented as

$$V_{i,j}(x_i, x_j) = E_{col}(x_i, x_j) + E_{curv}(x_i, x_j) \tag{9}$$

where E_{col} and E_{curv} are color similarity term and co-occurrence curvature term respectively.

2.3.3 The Color Similarity $E_{col}(x_i, x_j)$

When the proposals are similar, they are more likely to be combined to the same object. Otherwise, the combination should be punished. We define $E_{col}(x_i, x_j)$ as:

$$E_{col}(x_i, x_j) = 1 - exp(-\chi^2(h_{x_i}, h_{x_j})) \tag{10}$$

$$\chi^2(h_{x_i}, h_{x_j}) = \frac{1}{2}\sum_{b=1}^{N_d}\frac{(h_{x_i}(b) - h_{x_j}(b))^2}{h_{x_i}(b) + h_{x_j}(b)} \tag{11}$$

where h_{x_i} is the color histogram of proposal u_{x_i} selected by node v_i, N_d is the dimension length of the color histogram.

2.3.4 The Co-occurrence Curvature Term

The co-occurrence curvature term is based on the observation that a good combination of the segments leads to a smooth boundary. We tend to select the proposal pairs that makes the boundary of their combinational region smooth, which is denoted as:

$$E_{curv}(x_i, x_j) = curv(u_{x_i} \bigcap u_{x_j}) \tag{12}$$

2.4 Optimization by Dynamic Programming on a Tree

We next minimize the energy in Eq. (1). Inspired by the method in [16], the energy function is minimized by the dynamic programming technique on the MST structure. We first randomly select root node from MST structure since the minimization is robust to the random root selection. We denote the root vertex as v_r. For each node v_i except the leave nodes, the cost of assigning best label x_i^* consists of the cost of its children with the cost of assigning label x_i to the node, which can be represented by a recursive equation:

$$B_i(x_i) = D_i(x_i) + \sum_{v_j \in C_i} min_{x_j \in X}(B_j(x_j) + V_{ij}(x_i, x_j)) \tag{13}$$

where $B_i(x_i)$ is the value of storing the cost of the subtree from the leaf nodes to current node, X is the label set, and C_i denotes the children nodes of v_i. For each leaf node without children node, the cost can be depicted as $B_i(x_i) = D_i(x_i)$. Then, the cost of the whole MST tree is represented as $B_r(x_r)$, where x_r is the label of the root node v_r.

Finally, we trace the recursive equation back to find the global optimal solution:

$$x_i^* = argmin_{x_i \in X}(B_i(x_i) + V_{ij}(x_i, x_j)) \tag{14}$$

An minimization example is shown in Fig. 4, where we

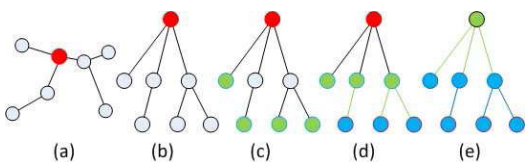

Fig. 4 The process of the dynamic programming on a tree. (a) The orginal MST. (b) The reshaped tree and the root node. (c) Calculating the assigning cost on the leaves nodes. (d) Expanding the subtrees by adding their parents recursively. (e) The final result by finishing the root node.

first select a node as the root node randomly, as the red node in Fig. 4 (a). Then, the original MST is reshaped to a general abstract tree based on the root node, as shown in Fig. 4 (b). Based on the general abstract tree, we next calculate the cost of assigning the labels to the leaves nodes as shown in Fig. 4 (c). Finally, the subtrees are expanded to the parent node, and the root node is labeled to get the global minimization as shown in Fig. 4 (d)-(e).

3. Experiment

In this section, classes *car* and *person* in Graz-02 dataset are used to verify the proposed method. Similar to [4], [7], we select the first 300 images, where 150 images are used as the training data and the rest images are used for verification.

In the parameter setting, we quantize the color histogram into 12 bins, i.e., $N_d = 12$, and set the number of the candidate object windows to $N = 100$. Moreover, in the initial proposal generation, we discard the proposals whose area is smaller than 1 percent of the window area.

We compare our method with the state-of-the-art algorithms, including [4], [7]–[10] and DPM+GrabCut. Because the result of [7] is a soft segmentation mask, we select the objective evaluation metric in [4], i.e., the F-measure $F = 2 \times rec \times pre/(rec + pre)$ calculated by the pixel-wise precision and recall. Table 1 shows the comparison results by the proposed method and the comparison methods.

From Table 1, we can see that the proposed method achieves the result of 61.5% in terms of the F-measure on *car* class, which outperforms the methods in [7], [8] and DPM+GrabCut. Furthermore, the performance of the proposed method is also better than the comparison methods [4], [7], [8], [10] and DPM+GrabCut on *person*. Meanwhile, the [9] has a best performance in the state of the art. They proposed a pylon model by hierarchical segmentation tree to get the semantic segmentation. Their methods use the pixel-wise labels as the groundtruth for the training set, while our method only needs the bounding boxes of the objects. In Fig. 5, we show some subjective results by the proposed method (in the bottom row) compared with the DPM+GrabCut method (in the second row). It is seen that the proposed method obtains better performance.

Table 1 The objective segmentation results in terms of F-measure (%).

method	car	person	average
Marszalek & Schmid [7]	53.8	44.1	49.0
Fulkerson et al. [8]	54.7	51.4	53.1
Aldavert et al. [10]	62.9	58.6	60.8
Lempitsky et al. [9]	83.7	84.9	84.3
Kuettle et al. [4]	74.8	66.4	70.6
DPM+GrabCut	61.3	54.3	57.8
Our Method	61.5	69.7	65.6

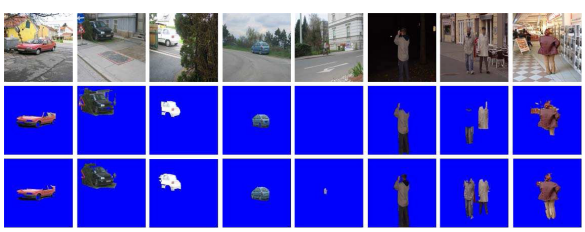

Fig. 5 The example results of the two subsets in Graz-02 dataset. The top row: the original images. The second and bottom rows: the results by DPM+GrabCut and proposed method, respectively.

4. Discussion

We also test our method on *bike*. The F-measure value is 43.0%, which is lower than the state-of-the-art methods. The first reason is that curvature and tuning points in the unary potential are not suitable for *bike*. Bicycle has fine structures and the segmentation of its parts is rough and not irregular, which results in a high potential. The second reason is that the segmentation candidates have the similar features with the background segments in the color space.

5. Conclusion

In this paper, we propose a full automatic and bounding box based foreground segmentation method. The minimum spanning tree is constructed to describe the spatial relationships and the dynamic programming is used to minimize the energy function. Without using the pixel-wise annotation, the proposed method makes use of bounding boxes in the training set for the target objects. Experimental results demonstrate the effectiveness of the proposed method.

References

[1] H. Li and K.N. Ngan, "A co-saliency model of image pairs," IEEE Trans. Image Process., vol.20, no.12, pp.3365–3375, Dec. 2011.
[2] H. Li, F. Meng, and K. Ngan, "Co-salient object detection from multiple images," IEEE Trans. Multimedia, vol.15, no.8, pp.1896–1909, 2013.
[3] F. Meng, H. Li, G. Liu, and K. Ngan, "Object co-segmentation based on shortest path algorithm and saliency model," IEEE Trans. Multimedia, vol.14, no.5, pp.1429–1441, 2012.
[4] D. Kuettel and V. Ferrari, "Figure-ground segmentation by transferring window masks," Proc. CVPR, pp.558–565, June 2012.
[5] Y. Boykov and M.P. Jolly, "Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images," Proc. ICCV, pp.105–112, 2001.
[6] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut": Interactive foreground extraction using iterated graph cuts," ACM Trans. Graph., vol.23, no.3, pp.309–314, Aug. 2004.
[7] M. Marszatek and C. Schmid, "Accurate object localization with shape masks," Proc. CVPR, pp.1–8, June 2007.
[8] B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," Proc. ECCV, pp.179–192, 2008.
[9] V.S. Lempitsky, A. Vedaldi, and A. Zisserman, "A pylon model for semantic segmentation," Proc. NIPS, pp.1485–1493, 2011.
[10] D. Aldavert, A. Ramisa, R. de Mantaras, and R. Toledo, "Fast and robust object segmentation with the integral linear classifier," Proc. CVPR, pp.1046–1053, June 2010.
[11] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," CVPR, pp.277–284, Sept. 2009.
[12] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models,"

IEEE Trans. Pattern Anal. Mach. Intell., vol.32, no.9, pp.1627–1645, 2010.

[13] T. Chan and L. Vese, "Active contours without edges," IEEE Trans. Image Process., vol.10, no.2, pp.266–277, Feb. 2001.

[14] F. Meng, H. Li, G. Liu, and K.N. Ngan, "Image cosegmentation by incorporating color reward strategy and active contour model," IEEE Trans. Cybern., vol.43, no.2, pp.725–737, April 2013.

[15] H. Li, F. Meng, B. Luo, and S. Zhu, "Repairing bad co-segmentation using its quality evaluation and segment propagation," IEEE Trans. Image Process., vol.23, no.8, pp.3545–3559, Aug. 2014.

[16] P. Felzenszwalb and R. Zabih, "Dynamic programming and graph algorithms in computer vision," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.4, pp.721–740, April 2011.