CONNECTIVITY SIMILARITY BASED TRANSDUCTIVE LEARNING FOR INTERACTIVE IMAGE SEGMENTATION

Yadong Mu, Bingfeng Zhou

Institute of Computer Science and Technology, Peking University, P.R. China
{muyadong,zhoubingfeng}@icst.pku.edu.cn

ABSTRACT

We propose a novel graph-based transductive learning approach for interactive image segmentation. Here the term "transductive" indicates a process that iteratively propagates information from user-labeled regions to unlabeled image pixels. For the application of interactive image segmentation, transductive approach has several advantages compared with traditional color probabilistic model based approach. However, previous transductive approaches for image segmentation usually utilize an 8-connected neighborhood system, which has low efficacy when transferring local information to remote pixels. The main contribution of this paper is to estimate pairwise pixel similarity based on a novel path-based metric (i.e. *connectivity similarity*), rather than local comparison with 8-connected neighbors. We further theoretically prove the computing complexity is on a polynomial order and provide convergence guarantee for the extra local smoothing operation that is introduced to further refine the initial results. Especially, the proposed method shows promising performance in the multi-label case. Various experiments are presented to illustrate its effectiveness.

Index Terms— interactive image segmentation, connectivity similarity, linear propagation

1. INTRODUCTION

Image segmentation is one of the traditional and important problems in computer vision and image processing. Its potential applications are especially wide, such as medical image analysis and personal photo editing. Fully automated image segmentation is possible yet prones to error, mainly because it is difficult to overcome the gap between local image features (e.g. colors, edges, textures) and high-level semantics. To enhance the segmentation quality, one extra reference image ([1],[2]) or flash/non-flash image pairs [3] can be introduced to provide extra useful hints, which proves effective yet complicates the image capturing process. Instead, in recent years semi-automated or user-aided image segmentation has attracted increasing interest, due to its low requirement and higher accuracy.

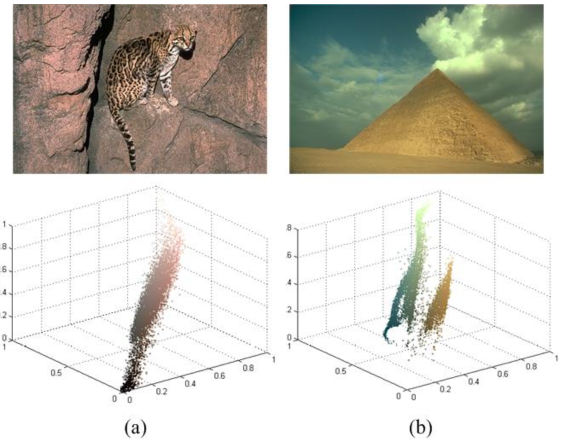

Fig. 1. Two difficult examples for inductive image segmentation. (a) color distributions of object/non-object are highly ambiguous. (b) heavy-tailed, asymmetric, uneven distribution which is distorted 2-manifold rather than 3D Gaussians and is difficult to be accurately represented by statistical models. See text for detailed explanation.

Generally speaking, most of popular interactive image segmenting approaches can be roughly divided into two categories: inductive or transductive way, which fundamentally differs in the way to utilize user guidance. In most inductive approaches, images are assumed to be drawn from certain statistical models (typically Gaussian Mixture Model, GMM), whose parameters can be optimally obtained via maximum likelihood or MAP estimation from seeds (i.e. user-labeled pixels). The most representative approaches are the *GrabCut* [4] system and its variant *LazySnapping* [5] developed by Microsoft. However, it is still open problem to find more reasonable models for image appearance. GMM, although simple and efficient, has several severe drawbacks. As illustrated in Figure 1-(a), when the desired objects and background share similar distribution in multi-dimensional color space, it is rather difficult to separate them based on GMM color model, since the information of spatial configuration is totally dropped during the model-training phase.

On the contrary, transductive graph-based methods [6] avoid explicitly feature modeling via non-parametric label propagation. Typically images are modeled as sparse graphs with 2D lattice topology. Individual pixel or overlapped small patches are treated as graph nodes, while adjacent pixels or patches are connected by an edge in the constructed graph. Assume ”seeds” have high confidence about their labels, and iteratively propagate it to remote unlabeled nodes along weighted graph edges. It is worthwhile to highlight two major issues in above process: graph construction (i.e. how to define pairwise similarity metric between pixels or patches), and graph propagation (i.e. transfer confidence to unknown image regions). Regarding the first issue, current approaches are mostly based on local comparison, efficient yet dropping all global information. Here we'll show that global method is possible.

In this paper we propose a novel interactive image segmentation method based on *connectivity similarity*. It consists of two steps: **graph construction** and **local smoothing**. We will discuss the details in Section 2 and here briefly list its advantages over other methods: first and most importantly, unlike conventional approaches, it performs graph construction with a robust, global, pairwise similarity definition rather than local ones. Regarding computing complexity, we present a proof of the existence of polynomial-time algorithm for similarity computation. Secondly, label smoothness between spatially nearby nodes is usually encouraged. A standard technique is the combination of Markov field modeling and min-cut-max-flow optimation [7]. However, for the NP-hard K-label segmentation ($K > 2$), variants of the min-cut-max-flow algorithm such as α-expansion Graph Cuts are especially time-consuming. Here we adopted an iterative *linear neighborhood propagation (LNP)* method for fast local smoothing. Theoretic analysis is provided to guarantee its convergence. Thirdly, the proposed method is especially convenient for user's retouching for initial segmentation results, and outperform many other existing algorithms in multi-label case (i.e. more than one desired objects to be cut out in an image.).

2. ALGORITHM

2.1. Confidence Estimation via Connectivity Similarity

As discussed in Section 1, we model images from a perspective of graph theory. Let $\mathcal{V} = \{x_i, i \in \mathcal{I}\}$ denote the vertex set for an image (each vertex corresponds to a pixel or *super-pixel* [8]), where $\mathcal{I}$ is the index set. In the context of interactive segmentation, typically we have the relations $\mathcal{I} = \mathcal{I}_{\mathcal{L}} \cup \mathcal{I}_{\mathcal{U}}$ and $\mathcal{I}_{\mathcal{L}} \cap \mathcal{I}_{\mathcal{U}} = \emptyset$ where $\mathcal{I}_{\mathcal{L}}$, $\mathcal{I}_{\mathcal{U}}$ are index sets for labeled and unlabeled nodes respectively. We connect two vertices if any of them is among the other's K-nearest-neighbors (K is an positive integer) in the spatial sense. By this means we build the edge set. Denote the constructed graph as $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ where $\mathcal{V}$, $\mathcal{E}$ represent sets of graph nodes and edges respectively. The simplest way for graph construction is treating each pixel as a single node, and define graph edges over 8-connected neighborhood, which is adopted in this paper but later we will discuss other candidates for acceleration purpose.

Estimation of edge weights in $\mathcal{G}$ is important yet not fully discussed in related literature. Without confusion, we will use both the term "distance" and "similarity" in the later sections, which are essentially equivalent under most situations. It is convenient to convert distance value to similarity value, and vice versa. Thus, we can measure edge weight either using distance d_{ij} for node i, j or similarity s_{ij}.

A standard method to calculate distance and similarity is the L_2 norm and *heat-kernel* similarity function, i.e.

$$d_{ij} = \parallel x_i - x_j \parallel^2,\ s_{ij} = \exp(-\frac{d_{ij}}{2\sigma^2}) \qquad (1)$$

However, for transductive segmentation, it is not a good choice for the case of few seeds provided and large number of remote nodes away from seeds, since the impact of seeds decays drastically when spatial distance becomes large. A distance metric defined basing on not only local distance but also global data distribution is able to overcome the above-mentioned problem. Here we adopt *connectivity similarity*, which is originally proposed by Fischer etc [9], where this idea is applied to data clustering. Other applications including face recognition and automated image segmentation can be found in [10]. However, to the best of our knowledge, there is no previous work about incorporating this idea into interactive segmentation.

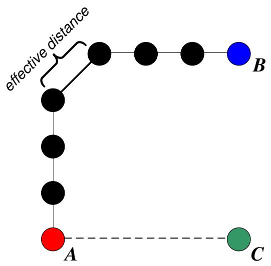

Fig. 2. Illustration for connectivity similarity. We have highlighted the effective distance between A and B.

The main idea behind connectivity similarity is to transform elongated structures to compact ones. As can be seen in Figure 2, according to Equation 1, unknown node A is more similar to source node C compared with B, since spatial distance $d_{AB} > d_{AC}$. However, it contradicts human intuition since there exists a path connecting A and B (in solid lines in Figure 2), i.e. A, B may lie on an *elongated manifold*. Formally, let us denote the collections of path in the graph from vertex i to j as $\mathcal{P}_{ij}$. In order to make two nodes connected by small-step path more similar, we define *effective distance* for each valid path $p \in \mathcal{P}_{ij}$ as the maximum step length, and the final distance between i, j is the minimum value of all effective distances among all $p \in \mathcal{P}_{ij}$, that is:

$$\hat{d}_{ij} = min_{p \in \mathcal{P}_{ij}} \left\{ \max_{1 \le h \le |p|-1} d_{p[h]p[h+1]} \right\} \tag{2}$$

$$\hat{s}_{ij} = \exp(-\frac{\hat{d}_{ij}}{2\sigma^2}) \tag{3}$$

Note that here we use different notations $\hat{d}$ and $\hat{s}$ from original d, s in Equation 1. See Figure 2 for an example of effective distance. In the context of interactive segmentation, denote the label set as $\mathcal{L} = \{1, 2, ..., L_{max}\}$ (note that our algorithm supports multi-label case, thus L_{max} can be greater than 2) and $L_i \in \mathcal{L}$ is the label of x_i. For unlabeled nodes, theirs labels are initialized as 0 (i.e. unknown). For each $i \in \mathcal{I}_{\mathcal{L}}$, we can calculate its connectivity distance to every unlabeled node in $\mathcal{I}_{\mathcal{U}}$. Finally, for each unlabeled node j, we obtain L_{max} distance values in all, each for a unique label in $\mathcal{L}$, which can be defined as:

$$d_j^l = \min_{i \in \mathcal{I}_{\mathcal{L}}, L_i = l} \hat{d}_{ij} \text{ and } s_j^l = \max_{i \in \mathcal{I}_{\mathcal{L}}, L_i = l} \hat{s}_{ij} \tag{4}$$

To illustrate the effect of connectivity similarity more intuitively, we present another example on the standard “two-spiral” dataset. In Figure 3, hundreds of points are randomly sampled from *positive* or *negative* class (the left subfigure), and 4 points per class are randomly selected and labeled as seeds (the right subfigure). Note that if calculating according to ordinary Euclidean distance, only those data points nearby the labeled ones have relatively small distance values, which ignores the fact that both classes actually lie on a thin, elongated manifold and inaccurate for classification task. For comparison, we plot the results obtained using *connectivity distance* in Figure 3 (the middle subfigure). It can be seen, two distinct points, even spatially far away from each other, seem “near” to the same class.

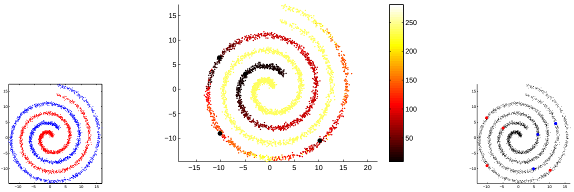

Fig. 3. **Left**: Groundtruth. **Right**: Partition of unlabeled nodes (in black) and Labeled nodes (positive class is in red, while negative class is in blue). **Middle**: illustration for connectivity distance calculated from the four labeled positive nodes. Note that remote nodes are also possible to have low distances. (Enlarge it for better viewing.)

One important issue about connectivity similarity is its computation complexity. Since the number of nodes in image graph is typically on the order of 10^4 or even higher, only polynomial-time algorithm with low degree is computationally acceptable. Based on graph theory, we can derive the following theorem:

Theorem 2.1. *There exists algorithm in polynomial time for the computation of connectivity similarity.*

Proof. The computation can be accomplished by slightly modifying classical Kruskal’s minimum spanning tree algorithm. In the beginning, edges in graph $\mathcal{G}$ are pushed into a stack in descant order according to their weights defined in Equation 1. Each pixel is initially treated as a single ”cluster” to be merged. Then the following process is iteratively performed until the stack is empty or $N - 1$ edges have been added into the spanning tree: pop up one element from the top of stack; denote its weight as d_{min}. If it link two nodes which are already in the same cluster, simply abandon it and run for the next iteration, otherwise it will act as the ”bridge” to connect two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ and should be added into the final spanning tree. Moreover, for any $p \in \mathcal{C}_i, q \in \mathcal{C}_j$, there is $\hat{d}_{pq} = d_{min}$, otherwise implying existence of another two clusters that can be merged by a ”bridge” with smaller weight. For complete input graph with N nodes the computing time is on the order of $\mathcal{O}(N^2 \log(N))$. □

2.2. Local Smoothing through Linear Propagation

After the computation described in Section 2.1, finally we get a $N \times L_{max}$ connectivity similarity matrix $\hat{\mathcal{S}}$, where $\hat{\mathcal{S}}(i, l) = s_i^l$ with definition in Equation 4. Each column vector of $\hat{\mathcal{S}}$ corresponds to all graph node’s confidence values for one specific label. However, the obtained similarity values are too noisy to directly perform graph node classification, as can be seen in Figure 5.

In MRF image modeling, smoothness assumption such as *Ising prior* is usually incorporated for outlier removal and local averaging. For the multi-label case, direct global optimization can be very time-consuming, even guided by heuristics, thus intolerant for real-time applications such as interactive segmentation. Instead we adopt a recently proposed local smoothing method named *linear neighborhood propagation* (*LNP*). The intuition behind *LNP* is that each graph node is able to iteratively improve its initial estimate by referring to the weighted averaged value of its neighbors. The key to LNP is to build the $N \times N$ sparse *propagation matrix* $\mathcal{W}$. The (i, j)-th entry in $\mathcal{W}$ is nonzero only when node j is among the k-nearest-neighbors of node i. Finally each row of $\mathcal{W}$ is normalized so that $\sum_j \mathcal{W}(i, j) = 1$.

The local smoothing operation proceeds according to the following formula:

$$\mathcal{Y}^{t+1} = \alpha \mathcal{W} \mathcal{Y}^t + (1 - \alpha) \hat{\mathcal{S}} \tag{5}$$

where $\alpha \in (0, 1)$ is a free parameter and $\mathcal{Y}^0$ is initialized as $\hat{\mathcal{S}}$. The term $(1 - \alpha)\hat{\mathcal{S}}$ is introduced to stay nearby their original values. Computation in Equation 5 is efficient due to $\mathcal{W}$'s high sparsity. Assume sequence $\{\mathcal{Y}^t\}$ converges to a stable point $\mathcal{Y}^*$, then the final label for node i can be simply determined via $L_i = \arg_l \max \mathcal{Y}^*(i, l)$. Also we have theoretic analysis for its convergence property:

Theorem 2.2. *The iterative updating procedure in Equation 5 will converge to a unique solution* $\mathcal{Y}^* = (1 - \alpha)(I - \alpha W)^{-1}\hat{\mathcal{S}}$*, where* I *is unit matrix in* $\mathcal{R}^{N \times N}$.

Proof. We omit the proof due to space limitation. □

3. EXPERIMENTS

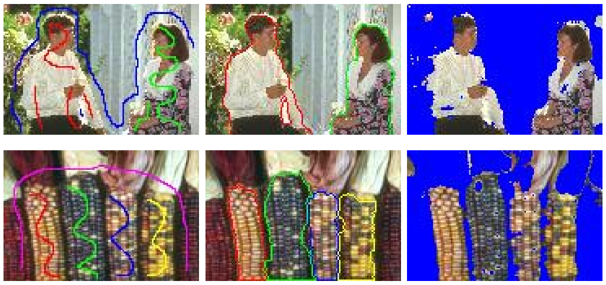

Fig. 4. Experimental results. **Left**: original image with multi-label user strokes. **Middle**: results with our proposed method. Contours of extracted objects are highlighted. **Right**: results with GMM based method.

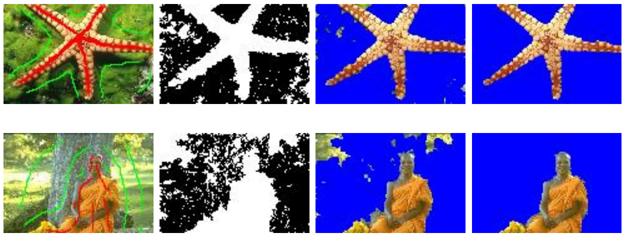

Fig. 5. The first column are original images with user strokes imposed on. The second column presents binarized results before local smoothing. And the last two columns are segmentation results after performing 1 and 20 times of local smoothing operations.

We evaluate the proposed method on the publicly available BSDS image dataset [11]. The experimental results is presented in Figure 5 and 4. As can be seen, the proposed method outperms traditional Gaussian mixture based method, especially in the multi-label case, and the LNP based local smoothing operation show its effectiveness to make the extracted object regions more consistent.

For image with large size, treating each pixel as a distinct node in the graph $\mathcal{G}$ will result in a huge graph, which indicates the above-mentioned procedure will be time-consuming. To overcome this difficulty, it is better to model a small adjacently-connected regions in the image as a basic graph node, i.e. we can first utilize techniques like super-pixel or watershed algorithm to over-segment the original image into thousands of small image patches, which results in a graph with a moderate number of nodes. We omit the super-pixel based experimental results due to space limitation.

4. CONCLUSION

We present a propagation-based interactive segmentation approach. It models image as graphs and globally estimates pairwise similarity via connectivity similarity, thus overcoming the problems of traditional local ones. Our method can provide comparable accuracy and computing speed compared with state-of-the-art ones. Various experiments on public BSDS300 image dataset prove its effectiveness.

5. REFERENCES

[1] Carsten Rother, Thomas P. Minka, Andrew Blake, and Vladimir Kolmogorov, “Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs.,” in *CVPR (1)*, 2006, pp. 993–1000.

[2] Yadong Mu and Bingfeng Zhou, “Co-segmentation of image pairs with quadratic global constraint in mrfs,” in *ACCV (2)*, 2007, pp. 837–846.

[3] Jian Sun, Sing-Bing Kang, Zongben Xu, Xiaoou Tang, and Heung-Yeung Shum, “Flash cut: Foreground extraction with flash/no-falsh image pairs.,” in *CVPR*, 2007.

[4] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, “”grabcut”: interactive foreground extraction using iterated graph cuts.,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[5] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum, “Lazy snapping.,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, 2004.

[6] Fei Wang, Changshui Zhang, Helen C. Shen, and Jingdong Wang, “Semi-supervised classification using linear neighborhood propagation,” in *CVPR (1)*, 2006, pp. 160–167.

[7] Yuri Boykov and Marie-Pierre Jolly, “Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images.,” in *ICCV*, 2001, pp. 105–112.

[8] Xiaofeng Ren and Jitendra Malik, “Learning a classification model for segmentation,” in *ICCV*, 2003, vol. 1, pp. 10–17.

[9] Bernd Fischer, Volker Roth, and Joachim M. Buhmann, “Clustering with the connectivity kernel,” in *NIPS*, 2003.

[10] Hong Chang and Dit-Yan Yeung, “Robust path-based spectral clustering with application to image segmentation,” in *ICCV*, 2005, pp. 278–285.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *ICCV*, July 2001, vol. 2, pp. 416–423.