Pattern Recognition 47 (2014) 2745–2755

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Interactive image segmentation via kernel propagation

Cheolkon Jung*, Meng Jian, Juan Liu, Licheng Jiao, Yanbo Shen

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 6 March 2013
Received in revised form
8 January 2014
Accepted 18 February 2014
Available online 27 February 2014

Keywords:
Interactive image segmentation
Kernel propagation
Pairwise constraints
Semi-supervised learning

ABSTRACT

In this paper, we propose a new approach to interactive image segmentation via kernel propagation (KP), called *KP Cut*. The key to success in interactive image segmentation is to preserve characteristics of the user's interactive input and maintain data-coherence effectively. To achieve this, we employ KP which is very effective in propagating the given supervised information into the entire data set. KP first learns a small-size seed-kernel matrix, and then propagates it into a large-size full-kernel matrix. It is based on a learned kernel, and thus can fit the given data better than a predefined kernel. Based on KP, we first generate a small-size seed-kernel matrix from the user's interactive input. Then, the seed-kernel matrix is propagated into the full-kernel matrix of the entire image. During the propagation, foreground objects are effectively segmented from background. Experimental results demonstrate that *KP Cut* effectively extracts foreground objects from background, and outperforms the state-of-the-art methods for interactive image segmentation.

1. Introduction

An important goal of image segmentation is to separate the desired foreground objects from the background. However, the color and texture features in natural images are generally very complex. Thus, the automatic segmentation of foreground objects from the complex background is a very difficult task [1–9]. Currently, semi-automatic segmentation methods incorporating simple user interaction have been actively studied [1,5–7,10,11]. Interactive image segmentation aims to extract foreground objects in complex scenes using the simple user interaction. In interactive image segmentation, the user's interactive information is effectively employed for getting some prior information which leads to good segmentation performance. Therefore, good performance in interactive image segmentation generally depends on the accurate segmentation of objects from the background and efficient handling of the interactive information [1,11].

1.1. Related work

Up to the present, a number of studies have been conducted to segment foreground objects using the user's interactive information [1,5–7,10–23]. Boykov and Jolly [14] proposed the interactive graph cuts based on *Graph Cut* to extract optimal boundaries and regions of objects. *Graph Cut* had been used to find the globally optimal segmentation of images, and provided good balance of boundary and region properties in image segmentation. Thus, the interactive graph cuts achieved robust segmentation even in images where color distributions of foreground and background are not well separated. Rother et al. [11] proposed *Grab Cut* which greatly improved the interactive graph cuts in terms of user interaction. To extract foreground objects, *Grab Cut* only needed simple interaction of dragging a rectangle around desired objects. In doing so, users indicated a region of background, and was free of any need to mark a foreground region. *Lazy Snapping* [15] was an interactive image cutout system which was easy to handle and produced high-quality foreground extraction results. *Lazy Snapping* was explicitly composed of two main parts: object context specification and boundary refinement. To reduce the computational complexity, *Lazy Snapping* combined *Graph Cut* with watershed segmentation. Hierarchical graph cuts [16] was also an extended method of *Graph Cut* which adopted the coarse-to-fine hierarchical strategy. The coarse-scale segmentation was performed on the basis of the initial interactive rectangle, and the accurate boundary extraction was conducted at the finer scale. The framework of the hierarchical graph cuts had pyramid structure to guarantee the robustness. Ning et al. [1] proposed maximal similarity based region merging (MSRM) using mean-shift segmentation. In MSRM, it was required that users marked the regions of foreground and background roughly by lines. Then, MSRM calculated the similarity of different regions and merged them based on the maximal similarity rule with the help of the markers. MSRM basically followed a two-step approach of over-segmentation and classification (i.e., split-and-merge strategy) which received much attention by many researchers. Consequently, the user's interactive information plays an

* Corresponding author.
E-mail address: zhengzk@xidian.edu.cn (C. Jung).

http://dx.doi.org/10.1016/j.patcog.2014.02.010
0031-3203

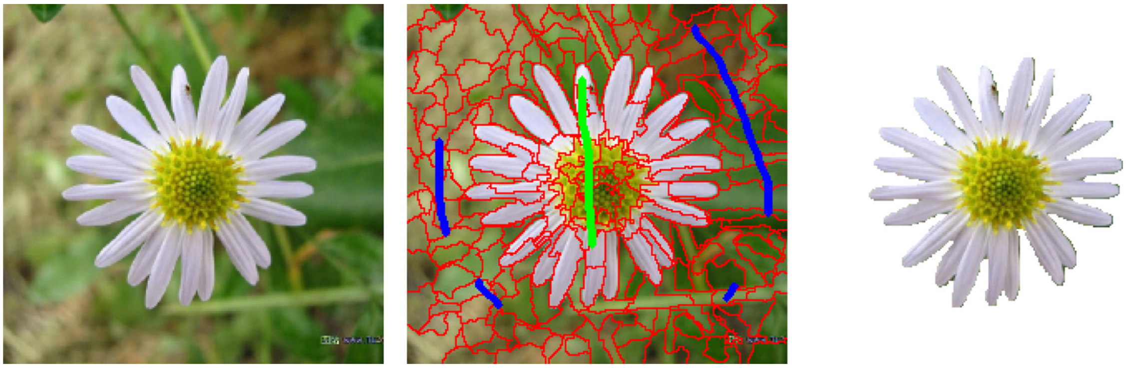

Fig. 1. Example of *KP Cut*. Left: original image. Middle: simple user interaction (green lines are foreground markers while blue lines are background ones). Right: foreground extraction results. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

important role in interactive image segmentation. To achieve good foreground extraction, further studies are needed to preserve characteristics of the user's interactive input and maintain data-coherence effectively.

1.2. Contributions

In this paper, we propose a new interactive image segmentation method based on kernel propagation (KP), called *KP Cut*. To preserve characteristics of the user's interactive input and maintain data-coherence effectively, we employ KP for interactive image segmentation. The KP method first learns a small-size seed-kernel matrix, and then propagates it into a large-size full-kernel matrix [24]. Because KP is based on kernel learning, it is very effective in propagating the user's interactive information into the entire image. First, *KP Cut* generates pairwise constraints using the user's interactive information. As shown in Fig. 1, users draw green and blue lines interactively on the segmented image by clicking and dragging with a mouse to mark foreground or background seeds, respectively. The superpixels, i.e. image segments, connected to green and blue markers are used to generate pairwise constraints of *must-link* and *cannot-link*, respectively. We use the *must-link* and *cannot-link* constraints as supervised information to learn the global discriminative kernel matrix for image segmentation. Second, *KP Cut* makes a small-size seed-kernel matrix using the pairwise constraints. Finally, the seed-kernel matrix is propagated into the full-kernel matrix of the entire image, and thus foreground objects are segmented. *KP Cut* is different from conventional interactive segmentation approaches in that it uses a learned kernel matrix from pairwise constraints for image segmentation. Therefore, experimental results demonstrate that *KP Cut* effectively extracts foreground objects from background by preserving interactive information and maintaining data-coherence in the learned kernel matrix. As a result, *KP Cut* outperforms state-of-the-art approaches for interactive image segmentation.

1.3. Organization

The rest of the paper is organized as follows: In Section 2, we briefly review KP which is the basis of this paper. In Section 3, we describe the proposed *KP Cut* for interactive image segmentation in detail. In Section 4, some experimental results and the corresponding analysis are provided. Finally, we draw conclusions in Section 5.

2. Kernel propagation (KP)

Kernel methods are effectively used in dealing with nonlinear problems in modern pattern recognition and machine learning. Recently, kernel learning is receiving much attention because a learned kernel can fit the given data better than a predefined kernel [24,25]. In cases where only inner products of the input data are involved, kernel learning is equivalent to kernel matrix learning (KML). KP is based on KML and inspired by label propagation (LP) [26] which propagates the seed labels from the labeled samples to the unlabeled samples. As shown in Fig. 2, if K_{ll} is a known seed-kernel matrix, KP aims to propagate K_{ll} to the other unknown sub-blocks K_{lu}, K_{ul}, and K_{uu}. Thus, the full-kernel matrix K is made to be known by KP. KP is formulated to be a minimization problem as follows:

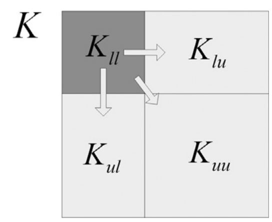

Fig. 2. Kernel propagation (KP), redrawn from [24]. An unknown full-kernel matrix K is split into four sub-blocks K_{ll}, K_{lu}, K_{ul}, K_{uu}. Given a seed-kernel matrix K_{ll}, KP aims to propagate K_{ll} into the other unknown blocks K_{lu}, K_{ul}, and K_{uu}; thus making the full-kernel matrix K to be known.

$$\min_{K\succeq 0} : \mathrm{Tr}(LK)$$
$$\text{s.t.}: \quad gKg^T = K_{ll}, \quad g=[I_l, 0] \tag{1}$$

where K is reorganized as $K=\begin{bmatrix}K_{ll} & K_{lu}\\ K_{lu}^T & K_{uu}\end{bmatrix}$, of which the sub-part $K_{ll}=gKg^T$ equals to a known seed-kernel matrix and I_l is the identity matrix of indexing the constrained samples. Because fewer constraints are imposed in (1), less computation is needed to solve the semi-definite programming problem (SDP) as compared to the pairwise constraint propagation (PCP) [25]. In (1), K has the decomposition of $K=BB^T$ due to its symmetry and positive semi-definiteness, where B is a real matrix. If we denote $B=\begin{bmatrix}B_l\\ B_u\end{bmatrix}$, $L=\begin{bmatrix}L_{ll} & L_{lu}\\ L_{ul} & L_{uu}\end{bmatrix}$, and $Q=\begin{bmatrix}I_l\\ -L_{uu}^{-1}L_{lu}^T\end{bmatrix}B_l$, the solution of (1) is obtained by the following equation:

$$K=BB^T=QB_lB_l^TQ^T=QK_{ll}Q^T$$
$$=\begin{bmatrix}K_{ll} & -K_{ll}L_{lu}L_{uu}^{-1}\\ -L_{uu}^{-1}L_{lu}^TK_{ll} & L_{uu}^{-1}L_{lu}^TK_{ll}L_{lu}L_{uu}^{-1}\end{bmatrix} \tag{2}$$

Eq. (2) is the closed-form solution of (1), and thus $\mathrm{Tr}(LK)$ can be reformulated as follows:

$$\mathrm{Tr}(LK)=\mathrm{Tr}[(L_{ll}-L_{lu}L_{uu}^{-1}L_{lu}^T)K_{ll}] \tag{3}$$

Thus, if we get the seed-kernel matrix K_{ll}, then we obtain the desired full-kernel matrix K. We denote $\mathbf{x}_l$ and $\mathbf{x}_u$ as constrained