Automatic Image Segmentation Incorporating Shape Priors via Graph Cuts

Xianpeng Lang\textsuperscript{1,2}, Feng Zhu\textsuperscript{1}, Yingming Hao\textsuperscript{1} and Qingxiao Wu\textsuperscript{1}
\textsuperscript{1}Shenyang Institute of Automation, Chinese Academy of Sciences
\textsuperscript{2}Graduate School of the Chinese Academy of Sciences
Shenyang, 110016, P.R.China
\{xplang,fzhu,ymhao,wuqingxiao\}@sia.cn

Abstract—In recent years, graph cut has been regarded as an effective discrete optimization method and received increasing attentions in vision community. However, many existing graph cut segmentation algorithms require interactive operations, which are not appropriate for automatic applications. In this paper, we propose an automatic segmentation algorithm via graph cut. Firstly, the data term in traditional graph cut energy is redefined to counteract illumination change. Secondly, shape priors are introduced into segmentation process, which help to obtain more robust results. Finally, an automatic segmentation strategy is presented. Experiments demonstrate that our segmentation algorithm can provide promising results, even when object suffering pixel intensity variation and continuously shape deformation.

I. INTRODUCTION

Image segmentation is a fundamental issue in computer vision, which aims at partitioning the image into several meaningful regions. During the past decades, a large amount of research work was performed, and subsequently various segmentation algorithms were presented. However, unconstrained segmentation is still an ill-posed problem. Background noise, occlusion, low contrast, and missing parts can yield artificial edges, discontinuous contours, or overlapping objects. Under these conditions, traditional general-purpose segmentation methods often tend to fail. Besides, altering of object’s pose and illumination can also lead to inconsistent segmentation results. In fact, these situations always happen in practical applications. Therefore, image segmentation is considered as a classic tough problem.

Low-level visual cues, such as intensity, color and texture, may be insufficient to make successful methods. Introducing shape priors to segmentation is proved to be a useful way of improving segmentation results. This kind of segmentation framework integrates low-level and high-level visual cues together, and then can increase the algorithm’s robustness against to noise, clutter, and occlusion. However, formulating shape information into segmentation framework still needs research. Furthermore, the incorporation of shape prior may raise computation complexity of algorithms, reducing their effectiveness under real-time applications.

Since the beginning of this century, many researchers have presented numerous algorithms to incorporate shape priors with active contour framework using variational methods, which demonstrated excellent performance. However, approximating numerical schemes of variational methods must be very carefully designed to insure robustness. In fact, variational method is numerical unstable and tends to trapped into local optimal solutions.

The research by D.M. Greig et al\cite{1} is known as the first one that combines graph cut with vision problem. They proposed that a graph can be build according to MAP-MRF energy function, and then one-one corresponding between MRF configurations and graph cuts was constructed. Therefore, the configuration with minimum energy can be found by the min-cut of the graph. Moreover, by Ford and Fulkerson’s theory\cite{2}, min-cut can be calculated by max-flow algorithm. Unfortunately, due to relative slow operation speed and application restriction, Greig’s work did not obtain deserved recognition for nearly ten years. This situation changed after Boykov et al’s\cite{3}\cite{4} development in graph cut. In recent years, graph cut has received increasing attentions in vision community. Under this framework, many low-level visual problems, e.g. segmentation, can be described as a pixel labeling problem. The image is described as a discrete MRF, and then MAP-MRF energy function is constructed, finally this energy is optimized by graph cut to find solutions. Graph cut segmentation is proved to guarantee global or strong local optima. Furthermore, it is also numerical stable and repeatable, namely, for same energy function one would always get identical segments despite of using which min-cut/max-flow algorithms.

However, many existing segmentation algorithms, which are based on graph cut, require user to provide information interactively. This operation makes these algorithms not appropriate for automatic applications. Incorporating shape priors into graph cut segmentation has been a new trend for these years. To our best knowledge, the shape priors can be divided into two groups: specific shape priors and generic shape priors. The former priors includes \cite{5, 6, 7}, which are designed to represent certain objects. In contrast, generic shape priors characterize shapes with some common features, which contain \cite{8} and \cite{9}.

In this paper, we address the aforementioned segmentation problems and present an automatic graph cut segmentation algorithm, which incorporating shape priors. The rest of this paper is organized as follows. In section 2, we review some preliminaries of graph cut segmentation. Then, we put
forward the proposed algorithm and explain its details in Section 3. Section 4 presents experiment results. Finally, conclusions are drawn in section 5.

II. PRELIMINARIES

In this section, we briefly outline the basic facts about graph cut segmentation method. For more details see [10] and [11].

A. Graph cut

Let \( G = (V, E) \) denote a directed weighted graph, which is composed of a set of nodes \( V \) and a set of directed edges \( E \) that connect them with nonnegative weights. The set \( V \) consists of two types of nodes: neighborhood nodes \( P \) which correspond to pixels, voxels, or other features, and two terminal nodes called \( s \) (source) and \( t \) (sink) that represent "object" and "background" labels. Normally, there are two types of edges in the graph: n-links and t-links. n-links connect pairs of neighboring pixels or voxels. Thus, they represent a neighborhood system in the image. The neighborhood system can be 4- or 8-connected for 2D images. The cost of n-links corresponds to a penalty for discontinuity between the pixels. t-links connect neighborhood nodes \( P \) with terminal nodes. The cost of a t-link connecting a pixel and a terminal corresponds to a penalty for assigning the corresponding label to the pixel.

A subset of edges \( C \subseteq E \) is called a cut, and the cost \(|C|\) of the cut is the sum of weights on these edges. All nodes are separated into two disjoint subsets \( S \) and \( T \), where \( s \in S, t \in T, S \subseteq V, \) and \( T \subset V \). There are no paths from terminal nodes \( s \) to terminal nodes \( t \) when all edges in the cut are removed. For a given graph, we can always find a cut that have the minimum cost, i.e. min-cut. According to Ford and Fulkerson’s work, min-cut was equivalent to maximum flow.

However, a fast implementation of graph cut algorithms can be a hard work in practice. Boykov and Kolmogorov[10] compare several graph cut algorithms and presented a new version of the max-flow algorithm that significantly outperformed the standard techniques. Furthermore, they also develop a set of max-flow C++ functions, which is available on their webpage[12]. In this paper, we utilize these codes for our experiments and the details are described in Section 4.

B. Energy minimization with graph cut

Image segmentation can be regarded as pixel labeling problems. Let \( L = \{l_1, l_2, \ldots, l_m\} \) be discrete label sets. In this paper, we consider a special label set, which contains only two labels: 0 and 1. Here 0 represent background pixel, while 1 represent object pixel. Consider a random field of images, in which discrete random variables \( X = \{x_1, x_2, \ldots, x_n\} \) are corresponding to image pixels. The elements in \( X \) can get values from label set \( L \), and for each \( x \), the probability of obtaining labels satisfies Markov property[13]:

\[
\Pr(x_u|\{x_v : u \in P - \{v\}\}) = \Pr(x_v|\{x_u : u \in N_v\}) \quad (1)
\]

where \( u, v \in P \) and \( N_v \) is aforementioned neighborhood system. If every element in \( X \) is assigned a label value, then we obtain a labeling configuration \( F = \{f_{x_1}, f_{x_2}, \ldots, f_{x_n}\} \) of random variables, where \( f_x \in L \). Then, image segmentation, i.e. labeling, can be regarded as a mapping \( F \) from \( X \) to \( L \). Given observe data \( D \), for each \( F \), we can compute posterior probability \( \Pr(F|D) \) for every configuration, which is also known as a MAP-MRF process. According to Bayes’ rule, the MAP-MRF energy function can be written as the following format,

\[
E(F) = \sum_{p \in X} D_p(f_p) + \sum_{p \in X, q \in N_p} V_{pq}(f_p, f_q) \quad (2)
\]

The energy function \( E \) is composed of two terms. The first term \( D_p \) is data term, which represents the penalties of assigning label \( f_p \) to pixel \( p \). The other term \( V_{pq} \) is interactive term, which punish the label disparities between neighboring pixels. In [13], \( V_{pq} \) is also mentioned as a smoothness energy. We can optimize this energy by graph cut method when \( V_{pq} \) is a submodule function[14]. Note that in this paper we focus on object/background segmentation with only two labels. Therefore, the energy is submodule when \( V(0,0) + V(1,1) \leq V(1,0) + V(0,1) \) is satisfied.

III. THE PROPOSED METHOD

In this section, we will present our automatic graph cut segmentation algorithm. At first, we give our energy formulation. Then, the shape prior is defined. After that is the segmentation algorithm. At last, the total processing flow is described.

A. Energy Formulation

The energy function used in our work is formulated as below:

\[
E(F) = \sum_{p \in H C} D_p(f_p) + \sum_{p \in X \cup N_p} V_{pq}(f_p, f_q) + \lambda E_{\text{shape}} \quad (3)
\]

Comparing with (2), this energy has two modifications. One is an additional shape prior term \( E_{\text{shape}} \), which will be discussed in next paragraph. \( \lambda \) is the coefficient that weights the importance of this shape term. The other is we only calculate \( D_p \) for hard constrains rather than all pixels in the graph. The symbol "HC" in (3) stands for "hard constrains".

Now let we explain the implements of \( D_p \) and \( V_{pq} \).

In [4], \( D_p \) consists of two kinds of pixels. One is hard constrain pixels which must be kept as object or background pixels. The other is soft constrain pixels, which are computed as the likelihood of observed data and color model. But in practical, the color model is hard to construct. Moreover, the color of objects may vary along with illumination change, background noise, etc. So in our work, we only preserve the \( D_p \) for hard constrain pixels and abandon it for other pixels. Specifically, our hard constrains are the centroid of object and a rectangle around it, representing \( s \) and \( t \), respectively.
Therefore, our modified intensities of internal region are homogeneous or continuous. Correct segmentations are always acquired, as long as the contrast between objects and background is strong or weak, the interactive terms. We can see that no matter the contrast by optimize the energy only with hard constrains and interactive terms. See text for more details.

We use the same \( V_{pq} \) as in [4], namely the Potts’ model, which can be express as follows:

\[
V_{pq}(f_p, f_q) = \exp \left( -\frac{(I_p - I_q)^2}{2\sigma^2} \right) \cdot \frac{1}{\text{dist}(p, q)} |f_p - f_q|
\]

(4)

where \( I_p \) and \( I_q \) denote the intensities of pixel \( p \) and \( q \), respectively. \( \text{dist}(p, q) \) is the Euclidean distance between pixel \( p \) and \( q \). \( \sigma \) is a positive value that can be estimated as "camera noise" [4]. It is easy to prove that (4) is a submodule function:

\[
V(0, 0) = V(1, 1) = 0, \text{ while } V(1, 0) \text{ and } V(0, 1) \text{ is non-negative, so that } V(0, 0) + V(1, 1) \leq V(1, 0) + V(0, 1).
\]

Fig.1 shows two segmentation examples, which obtained by optimize the energy only with hard constrains and interactive terms. We can see that no matter the contrast between objects and background is strong or weak, the correct segmentations are always acquired, as long as the intensities of internal region are homogeneous or continuous. Therefore, our modified \( D_p \) is not sensitive to illumination change.

B. Our Shape Prior

In order to further improve robustness, we incorporate shape priors into our segmentation energy. The shape prior we used is a level set representation of shape. Fig. 2 shows a human and its shape prior, where brighter intensities denote higher values. We add this value to t-links, so if a pixel is inside the shape then it tends to get high t-link weight. Furthermore, in order to make our algorithm process images faster, we can also use the binary image as the shape priors directly.

Note that in [6] the author also used level set to represent shape priors, but they modified n-links of the graph which is not the same as our work. Another important point need to notice is that we only adjust the value of t-links, hence n-links remains unchanged, and that is, the submodule property of segmentation energy is not disturbed. Therefore, if (2) can be optimized by graph cut, then energy function (3) can either. Remember that the Potts’ model we used is proved to be a submodule function in the previous paragraph, so we certainly can optimize (3) by graph cut.

C. Automatic Segmentation Strategy

Up to now, we have explained our segmentation energy in detail. The succeeding work is to find a strategy to make segmentation an automatic procedure.

The first thing need to resolve is acquiring shape prior automatically. Because our object may slightly change its shape during segmentation course, an updating operation is needed. In our work, we employ the segmentation result of previous frame as the shape prior of current frame.

Once the proper shape prior is obtained, we start to build the graph with respect to our segmentation energy in (3). In this step, weight values of t-links and n-links are computed. Remember that the t-links contains only hard constrains and shape priors. For object hard constrains, we use the centroid of previous segmentation results, while an external rectangle of previous results for background hard constrains. At last, shape priors are added to t-links.

After graph construction, the efficient graph cut algorithm in [10] is used to get the optimal segmentation result. Such segmentation result will be utilized in next frame. Therefore, we find an automatic segmentation strategy.

D. Processing Flow

The algorithm processing flow is summarized as follows:

1) Give initial centroid and size
   Construct graph
   Run graph cut and get initial segmentation result
2) Get information from previous segmentation result: shape, centroid, and size
3) Compute shape representation of shape prior
4) Construct graph
5) Run graph cut and get current segmentation result
6) Repeat Step 2 to Step 5 until all images are processed.

IV. EXPERIMENTS

We test our algorithm on the CAVIAR[15] image sequences. The program is coded with VC++ 2005 and run on a computer with Inter 2.83GHz CPU and 2GB RAM. We also compared our algorithm with the original one in which t-links include soft constrains instead of shape priors.

Fig.3 illustrates how we automatically obtain shape prior and hard constrains. Topleft is the segmentation results of previous frame, from which the object centroid and external...
rectangle are determined as the hard constrains and the shape priors is obtained as well. Here we assume the movement of the object is not sudden and quick, that is, no more than five pixels between two frames. These information is used in current frame shown in topright to build the corresponding graph for segmentation energy defined as in (3), and by graph cut we get the current segmentation result shown in bottomleft. Afterwards, we update the hard constrains and shape prior described in bottomright for next frame.

The segmentation results of two frames in image sequence are demonstrated in Fig.4. The top two images are original image frames, our segmentation object is the person on the rightside. The middle two images are segmentation results by original algorithms. Due to the soft constrains derived from pixel color model, many regions contain similar intensities are also regarded as the object. Without shape prior, the results were disturbed by other person (mid-left) and background clutter (mid-right). The bottom two images are our segmentation results. It is clear that the correct results are obtained by the proposed algorithm, which is not sensitive to illumination change and make significantly improvements with segmentation robustness.

V. CONCLUSIONS AND FUTURE WORKS

In recent years, graph cut has been regarded as an effective discrete optimization method and received increasing attentions in vision community. However, many existing segmentation algorithms based on graph cut require interactive operations, which make these algorithms not appropriate for automatic applications. In this paper, we proposed an automatic segmentation algorithm and incorporates shape priors to improve robustness. Experiments present promising results, even when object suffering pixel intensity variation and continuously shape deformation. Although the proposed method is able to segment continuous frames, the first frame still need warm-up operation. Moreover, this method cannot deal with very fast moving objects. Our future work will focus on automatic start-up and adding movement prediction to overcome these deficiencies.

REFERENCES