Constrained Active Contours for Boundary Refinement in Interactive Image Segmentation

Nguyen Thi Nhat Anh, Jianfei Cai, Juyong Zhang, Jianmin Zheng
School of Computer Engineering, Nanyang Technological University
Email: {ngnanh, asjfcai, S070051, asjmzheng}@ntu.edu.sg

Abstract—The state-of-the-art interactive image segmentation algorithms are often not able to produce accurate segmentation results with one-shot user input, and they frequently rely on laborious user editing to refine the segmentation boundary. In this paper, we propose a constrained active contour method for boundary refinement, which can be used to improve the segmentation results of many existing region-based interactive segmentation algorithms. Our constrained active contour model exhibits many desired properties for a good boundary refinement tool, including the robustness to user inputs, the ability to produce a smooth and accurate boundary contour, and the ability to handle topology changes. Experimental results show that the proposed refinement tool is highly effective and can significantly improve initial segmentation results without additional user inputs.

I. INTRODUCTION

Interactive image segmentation, which incorporates little user interaction to define the desired content to be extracted, has received much attention in the recent years. Many interactive image segmentation algorithms have been proposed in the literature. Most recent interactive image segmentation algorithms take the regional information as the input, where the user draws two types of strokes to label some pixels as foreground or background pixels and the algorithm completes the labelling for other pixels. The recent state-of-the-art region-based interactive segmentation algorithms include the graphcut-based methods [2], [8], the Random Walks based methods [6], [12], [11], the Geodesic methods [1], [5], etc.

Despite the performance improvement, the existing region-based approaches are still far away from achieving robust and accurate segmentation. One major problem is that the cutting contours generated by the existing region-based approaches, especially by Random Walks and Geodesic, are usually jaggy and do not snap to geometry features (see the first row of Fig. 1). Another problem is that the existing region-based approaches are sensitive to the user input in terms of either the number of seeds (the graphcut based algorithms) or the locations of seeds (the Random Walks and Geodesic algorithms), as pointed out in [9].

Thus, additional refinement step is often needed to improve the segmentation performance of the existing region-based methods. Most of the state-of-the-art interactive image segmentation methods [8], [6], [1], [11] rely on additional user inputs to either globally or locally refine the cutting contour. However, when dealing with complex images, the user is often required to provide a lot of additional strokes or boundary points and thus struggles with laborious refinement/editing. Another way for boundary refinement is to use the active contours/Snakes model [7] to refine the initial boundary contour produced by a region-based segmentation approach as in [4]. However, the refinement based on Snakes is only able to change the contour locally for smoothness but incapable of evolving the entire contour to snap to geometry features/edges and incapable of handling topology changes of the evolving contour.

Therefore, it is highly desired to have a “strong” boundary refinement method, which can evolve the contour entirely to absorb the non-robustness of region-based approaches and keep the contour smooth and snapping to geometry features. In this paper, we propose a constrained active contour method for boundary refinement which can be combined with many existing interactive segmentation methods to improve their segmentation results without additional user inputs (see the second row of Fig. 1). Our constrained active contour method is based on the convex active contour model [3], which makes use of both the boundary and the regional information to find an “optimal” solution.

The rest of the paper is organized as follows. Section II presents our proposed constrained active contour model, which incorporates the user input and the initial segmentation result into the convex active contour model. Section III shows the experimental results to demonstrate the effectiveness of our proposed method. Finally, section IV concludes the paper.

II. PROPOSED CONSTRAINED ACTIVE CONTOUR

The proposed constrained active contour method is based on the convex active contour model proposed in [3], which can be generally expressed as

$$\min_{0 \leq u \leq 1} \left( \int_{\Omega} g_u |\nabla u| dx + \lambda \int_{\Omega} h_r u dx \right),$$  \hspace{1cm} (1)

where $u$ is a function on image domain $\Omega$ and receives a value between 0 and 1 at each pixel location $x$ in the image, function $g_u$ is typically an edge detection function, and function $h_r$ is a region function that measures the inside and outside regions.

Basically, Eq. (1) consists of two terms balanced by a tradeoff factor $\lambda$, where the first term is a boundary term and the second term is a region term. The boundary term favors the segmentation along the curves that the edge detection function reaches minimum and also favors the segmentation with smooth boundary curve. The second term ensures the
Fig. 1. The segmentation results of different algorithms, including Geodesic [1], Random Walks [6] and GrabCut [8], and the corresponding refinements using our proposed method. The existing three algorithms produce jaggy boundary contours while our constrained active contour method is able to smooth out the contours and make them snap to geometric edges without any additional user input.

segmentation complying with some region coherence criteria defined in function $h_r$. Once the optimization problem of (1) is solved, the segmented region is found by thresholding the function $u$, i.e. $\{ x | u(x) > T \}$ (by default $T = 0.5$).

In the following, we discuss how to extend the convex active contour model of (1) that is designed for automatic image segmentation for refining the boundary obtained by a region-based interactive segmentation algorithm.

A. Contour Initialization

Let the result of an existing interactive image segmentation approach be denoted by a probability map $P(x)$, whose value is within the range of [0,1] indicating the probability that pixel $x$ belongs to the foreground region. For some segmentation methods like Random Walks [6] or Geodesic [1], this probability map, which is also known as alpha map, is directly produced as the result of the segmentation process.

For other segmentation methods like GrabCut [8], there is only the hard segmentation map but no such probability map produced. In this case, we produce the probability map based on the hard segmentation map and the foreground/background Gaussian mixture models (GMM). In particular, let $H(x)$ denote the hard segmentation map, where a value of 0 or 1 indicates a background or foreground pixel respectively. Considering the areas far away from the boundary are likely to be correctly classified, we initialize the probability map $P(x)$ as

$$P(x) = \begin{cases} 1, & \text{if } d(x) \geq D \text{ and } H(x) = 1 \\ 0.5 + 0.5(d/D)^2, & \text{if } d(x) < D \text{ and } H(x) = 1 \\ 0.5 - 0.5(d/D)^2, & \text{if } d(x) < D \text{ and } H(x) = 0 \\ 0, & \text{if } d(x) \geq D \text{ and } H(x) = 0 \end{cases}$$

(2)

where $d(x)$ denotes the Euclidean distance from pixel $x$ to its nearest boundary point, and $D$, a threshold value on $d$, is empirically set to 8.

Then, we check the hard segmentation map $H(x)$ against the foreground/background Gaussian mixture models (GMM) for consistency. Foreground/background GMMs introduced in [10] are estimated from foreground/background seeds and used to represent the color distributions of the foreground and background regions. Specifically, let $Pr(x|F)$ and $Pr(x|B)$ denote the probabilities that pixel $x$ fits the foreground and background GMMs, respectively. The normalized likelihood that $x$ belongs to foreground is $Pr(x) = \frac{Pr(x|F)}{Pr(x|F) + Pr(x|B)}$.

For a pixel $x$, if it belongs to either the case of $Pr(x|F) < 0.4$ and $H(x) = 1$ or the case of $Pr(x|F) > 0.6$ and $H(x) = 0$, we consider the pixel as an ambiguous one and reassign its $P(x)$ to equal to 0.5.

Once the probability map $P(x)$ is available, we initialize the contour evolution in (1) by assigning $P(x)$ to the function $u(x)$.

B. Regional Term Formulation

The foreground and background strokes give an excellent description about the color distributions of the foreground and background regions, which can be modelled by foreground/background GMMs, as discussed above. We incorporate this regional information into the regional term of the convex active contour model as

$$h_r(x) = Pr_B(x) - Pr_F(x) = \frac{Pr_F(x|B) - Pr_F(x|F)}{Pr_F(x|F) + Pr_F(x|B)}$$

(3)

where $Pr_F(x)$ and $Pr_B(x)$ are the normalized probabilities for the foreground and the background respectively.

This definition of $h_r$ ensures that the active contour evolves towards the one complying with the known GMM models. For instance, for a pixel $x$, if $Pr_B(x) > Pr_F(x)$ (resp. $Pr_B(x) < Pr_F(x)$) and $Pr_B(x) - Pr_F(x)$ is positive (resp. negative), $u(x)$ tends to decrease (resp. increase) during the contour evolution in order to minimize (1), which can lead to $u(x) \leq T$ (resp. $u(x) > T$) and the classification of the pixel belonging to the background (resp. the foreground).

The $h_r$ definition of (3) fails in the case that the foreground and background color models are not well separated. Thus, to avoid this problem and also make use of the segmentation
result obtained by the initial segmentation discussed in section II-A, we further propose to incorporate the probability map $P(x)$ into the regional term $h_r$ as

$$h_r(x) = \alpha(P_B(x) - P_F(x)) + (1 - \alpha)(1 - 2P(x))$$  \hspace{1cm} (4)

where $\alpha$, $\alpha \in [0,1]$, is a tradeoff factor. The second term $(1 - 2P(x))$ in (4) prevents the refined contour drifting too far apart from the initial segmentation. Specifically, when $P(x) > 0.5$ and $(1 - 2P(x))$ is negative, $u(x)$ tends to increase in order to minimize (1), which favors classifying the pixel as a foreground pixel, vice versa.

It is important to properly set the tradeoff factor $\alpha$ in (4). When the foreground and background colors are well separable, it is desired that the first term in (4) becomes dominating; otherwise, the second term in (4) should dominate. Thus, similar to the one suggested in [11], we set $\alpha$ according to the distance between the foreground and the background GMMs.

In addition to incorporating the probability map into the regional term, we further enforce some hard constraints in the contour evolution process. In particular, for those pixels that have no ambiguity in classification, including the pixels lying on the foreground/background strokes and the pixels having very large or very small $P(x)$ values, we treat them as hard constraints in the contour evolution process. We directly assign a negative $h_r$ value and a positive $h_r$ value, both with extremely large magnitude, to these confirmed foreground and background pixels, respectively. In this way, we guarantee that the refined result complying with the user input and also exploit more information from the initial segmentation result.

C. Boundary Term Formulation

The boundary term of $\int_{\Omega} g_b(x) |\nabla u| dx$ in (1) is essentially a weighted total variation of function $u$, where the weight $g_b$ plays an important role. $g_b$ is often an edge detection function such as

$$g_b(x) = \frac{1}{1 + |\nabla I(x)|^2}$$  \hspace{1cm} (5)

where $I(x)$ is the intensity of image pixel $x$. This definition is effective in the sense that it encourages the segmentation along the curves where the edge detection function is minimal. The problem with (5) is that at the locations with weak edges the boundary is likely to be smoothed out. Thus, in this paper, we propose to incorporate the GMM probability map $P_F(x)$ to enhance the edge detection. Particularly, we define $g_b$ as

$$g_b = \beta \cdot g_e + (1 - \beta) \cdot g_c$$  \hspace{1cm} (6)

where $g_e$ and $g_c$ are the results of applying the edge detection to the GMM probability map $P_F(x)$ and the original image, respectively, and $\beta, \beta \in [0,1]$, is a tradeoff factor computed in a similar way as $\alpha$. Note that the edge detection function returns values between 0 to 1 and a small value of $g_b$ corresponds to a likely edge.
Fig. 2. The segmentation results of different algorithms, including Geodesic [1], Random Walks [6] and GrabCut [8], and the corresponding refinements using our proposed method. First column: input images and user strokes; second column: probability maps; third column: the original segmentation results; fourth column: the refined segmentation results using our boundary refinement method.

Fig. 3. The segmentation results with different user inputs.