Robust Interactive Image Segmentation Using Convex Active Contours

Thi Nhat Anh Nguyen, Jianfei Cai, *Senior Member, IEEE,* Juyong Zhang, and Jianmin Zheng

Abstract—The state-of-the-art interactive image segmentation algorithms are sensitive to the user inputs and often unable to produce an accurate boundary with a small amount of user interaction. They frequently rely on laborious user editing to refine the segmentation boundary. In this paper, we propose a robust and accurate interactive method based on the recently developed continuous-domain convex active contour model. The proposed method exhibits many desirable properties of an effective interactive image segmentation algorithm, including robustness to user inputs and different initializations, the ability to produce a smooth and accurate boundary contour, and the ability to handle topology changes. Experimental results on a benchmark data set show that the proposed tool is highly effective and outperforms the state-of-the-art interactive image segmentation algorithms.

Index Terms—Interactive image segmentation, convex active contour, digital image editing.

I. Introduction

Interactive image segmentation, which incorporates small amount of 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. In general, interactive image segmentation algorithms can be classified into two categories: boundary-based approaches and region-based approaches.

In boundary-based approaches, the user is often asked to specify an initial area that is close to the desirable boundary. The active contours/Snake method [1] attempts to evolve an initial contour toward the object boundary. Methods based upon intelligent scissors [2], [3] apply Dijkstras shortest path algorithm to find a path between boundary seed points specified by the user.

Considering that the boundary-based approaches require great care to specify the boundary area or the boundary points, especially for complex shapes, most recent interactive image segmentation algorithms take the regional information as the input. In particular, in region-based approaches, the user is often asked to draw two types of strokes to label some pixels as foreground or background, after which the algorithm completes the labelling of all other pixels. State-of-the-art region-based interactive segmentation algorithms include Graph Cut based methods [4], [5], Random Walks based methods [6], [7], [8], and Geodesic methods [9], [10]. All these methods basically treat an image as a weighted graph with nodes corresponding to pixels in the image and edges being placed between neighboring pixels, and minimize a certain energy function on this graph to produce a segmentation.

In this paper, we consider the problem of interactive image segmentation with the input of foreground and background strokes, which requires only a small amount of interaction from the user. By carefully examining the state-of-the-art region-based approaches, we find that their performance is limited in terms of robustness and accuracy. First, many of the state-of-the-art region-based methods are overly sensitive to small variations in the interactions provided by the user (see Fig. 1). As pointed out in [11], the Graph Cut algorithm is sensitive to the number of seeds, while the Random Walks and Geodesic algorithms are sensitive to locations of seeds. This is mainly due to the different behaviors of the different energy functions. For example, Graph Cut tries to minimize the total edge weights along the cut. Thus, it may return very small segmentations (known as the "small cut" problem) in the case with small number of seeds provided. Random Walks based approaches minimize a Dirichlet energy functional with respect to the boundary conditions (locations of seeds), and variations in the boundary conditions result in different harmonic functions.

Second, the boundaries generated by the region-based approaches, especially those generated by Random Walks and Geodesics based approaches, are often jaggy and do not adhere to the geometric features in the image (see Fig. 1). An 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 [5], [6], [9], [8] rely on additional user inputs to either globally or locally refine the boundary. 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 [1] to refine the initial boundary contour produced by a region-based segmentation approach as in [12]. 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.
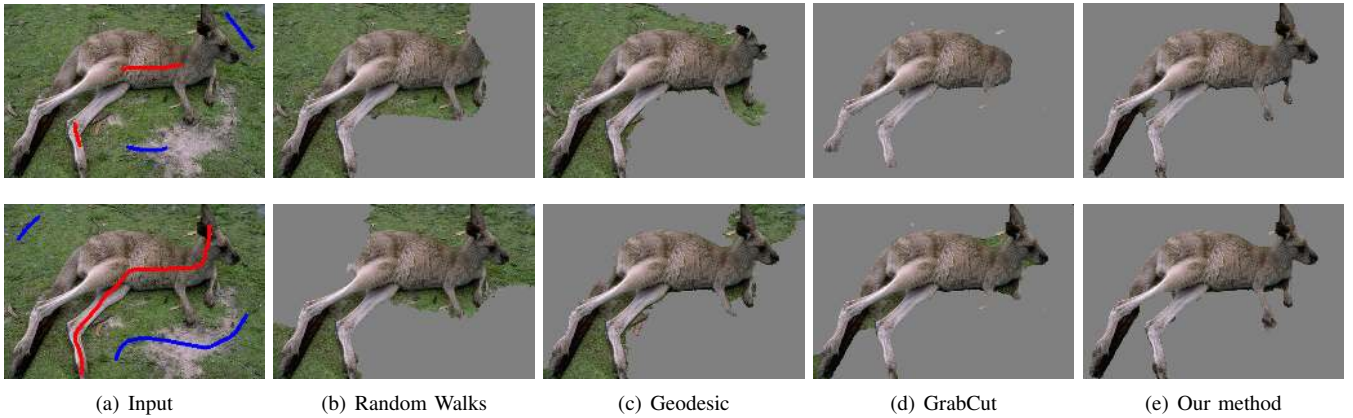
Fig. 1. The segmentation results of different algorithms, including Random Walks [6], Geodesic [9] and GrabCut [5], and our proposed method. Note that GrabCut is an advanced version of Graph Cut. The three state-of-the-art algorithms are overly sensitive to different user inputs while our method is quite robust. In addition, the existing methods 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 additional user input.

The above observations motivate us to design a new method for interactive image segmentation. The mathematical tool at the heart of the new method is the continuous-domain convex active contour model [13], which makes use of both the boundary and the regional information to find a global "optimal" solution. Continuous-domain convex methods have started to receive attention since they avoid the inherent grid bias in all discrete graph-based methods and also have fast and global numerical solvers through convex optimization [13], [14]. However, the convex active contour model so far has mainly been applied for automatic image segmentation, which often results in over-segmentation with trivial solutions for complex images [13], [15]. On the other hand, it is not clear how to apply the convex active contour model [13] for interactive image segmentation. Directly incorporating user inputs as hard constraints into the model does not lead to better performance. The major contributions of this paper include

- We propose to marry the powerful continuous-domain convex active contour with one of the state-of-the-art region-based methods, either Geodesic or Random Walks (Geodesic is chosen due to its fast processing speed), where the region-based method is used in the first step to generate an initial contour and the convex active contour is then applied in the second step to optimize the contour. Note that here we not only use the region-based method to generate an initial contour, but also incorporate the information obtained in the pre-segmentation into the convex active contour model, which is non-trivial. Such an integration utilizes the seed propagation and the location features introduced by Geodesic / Random Walks to reduce the possible "small cut" problem in the convex active contour, and also the powerful contour evolving capability provided by the convex active contour model to absorb the non-robustness of the region-based approaches. It ensures that the contour evolving does not drift too far away from the initial contour, complies with the user input, reflects the user intention and snaps the contour to geometry features.
- In addition, considering that the convergence speed for solving the convex active contour model is generally slow,

we make use of the Split Bregman method, as in [16], to solve the proposed constrained convex active contour model rapidly.

Experimental results show that the proposed method is fast, robust to different user inputs and different initializations, and capable of producing good results reflecting user intention and geometric features.

The rest of the paper is organized as follows. Section II reviews the general convex active contour model designed for automatic image segmentation and the use of the Split Bregman method to solve the model. Section III presents our proposed constrained active contour model, which incorporates the user input and the initial segmentation result into the convex active contour model. Section IV shows the experimental results to demonstrate the effectiveness of our proposed method. Finally, Section V concludes the paper.

II. RELATED WORK

As previously noted, the classical active contour model [1] is primarily used to perform local contour adjustment to improve the smoothness. The geodesic active contour model proposed in [17] is capable of evolving the entire boundary contour to snap to geometry edges, but it heavily depends on the edge detection function. The active contour without edges model [18] adds in regional information and removes the dependency on edge detection, but it is often trapped in local minimum due to non-convex modelling. In this section, we briefly review the convex active contour model recently introduced in [13], which is able to find the global minimum solution, and its application on automatic image segmentation. Furthermore, we also summarize the use of the Split Bregman method for solving the convex active contour model rapidly, which has been discussed in [16].

A. Convex Active Contour Model

The convex active contour model introduced in [13] can be generally expressed as

$$\min_{0 \le u \le 1} (\int_{\Omega} g_b |\nabla u| \ dx + \lambda \int_{\Omega} h_r u \ dx), \qquad (1)$$

with the following symbol definitions:

- u is a function on image domain Ω, which receives a value between 0 and 1 at each pixel location x in the image. The segmented region is obtained by thresholding the function u.
- Function g_b is a boundary function, which is often an edge detection function such as

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

  where $I(x)$ is the intensity of image pixel x.
- Function h_r is a region function that measures the inside and outside regions. Particularly, $h_r = h_r^{in} - h_r^{out}$, where h_r^{in} and h_r^{out} are the inside and outside region functions, respectively. They are often defined as

$$h_r^{in}(C_{in}, x) = (\mu_{in} - I(x))^2, \ \mu_{in} = \frac{\int_{C_{in}} I(x) \ dx}{\int_{C_{in}} dx}$$

$$h_r^{out}(C_{out}, x) = (\mu_{out} - I(x))^2, \ \mu_{out} = \frac{\int_{C_{out}} I(x) \ dx}{\int_{C_{out}} dx},$$

  where μ_{in} and μ_{out} are the mean intensities for inside and outside regions, C_{in} and C_{out}, respectively.

Basically, Eq. (1) consists of two terms balanced by a tradeoff factor λ, where the first term is a boundary term and the second term is a region term. The boundary term favors segmentations with boundaries along areas where the gradient is small, and also favors segmentations which have a smooth boundary. The second term ensures the 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.

$$C_{in} = \{x | u(x) > T\}, \qquad (3)$$

where typically $T = 0.5$.

The automatic segmentation problem based on the convex active contour model of (1) is usually solved by an alternate iterative approach depicted as follows.

1) Fix the segmentation, i.e. C_{in} and C_{out}, and update h_r.
2) Fix h_r to find the solution u for (1).
3) Update C_{in} and C_{out} according to (3)

The above three steps are repeated until convergence (i.e. when C_{in} and C_{out} no longer change). It can be seen that the computation bottleneck of this iterative approach lies in step 2, i.e. solving the optimization problem of (1).

B. Split Bregman Solver

Several methods have been proposed to solve (1) for a given h_r. Chan et al. [15] proposed to either enforce the inequality constraint of ($0 \leq u \leq 1$) using an exact penalty function, which is non-differentiable, or regularize the penalty function, which does not exactly enforce the inequality constraint. Bresson et al. [13] used a splitting/regularization approach to minimize (1). Their method "smears" the values of u near the object boundaries, and thus makes the segmentation results more dependent on the cutoff parameter T, which could eliminate the segmentation details.

Recently, Goldstein et al. [16] proposed to use the Split Bregman method to solve (1). The Split Bregman method is not only able to solve the convex active contour model but also a much more efficient solver. In the following, we briefly summarize the use of this Split Bregman solver as introduced in [16].

In particular, instead of solving (1) directly, Bregman introduced a new vectorial function d into the model as

$$\min_{u \in [0,1]} \int_\Omega g_b |d| + \lambda h_r u \ dx, \qquad (4)$$

with the constraint of $d = \nabla u$. This constraint is enforced using the efficient Bregman iteration approach defined as

$$\begin{aligned} (u^{k+1}, d^{k+1}) &= \arg \min_{u \in [0,1], d} \int_\Omega g_b |d| + \lambda h_r u \\ &\quad + \frac{\mu}{2} |d - \nabla u - b^k|^2 \ dx \qquad (5) \\ b^{k+1} &= b^k + \nabla u^{k+1} - d^{k+1}, \qquad (6) \end{aligned}$$

where k is the iteration index, $k \geq 0$. The computations of (5) and (6) are repeated until convergence.

Since (5) is differentiable, it can be solved using a simple alternating method. Specifically, the function is first differentiated with respect to u using the Euler-Lagrange Differential Equation, which results in the following optimality condition for u:

$$\mu \Delta u = \lambda h_r + \mu \operatorname{div}(d^k - b^k), \ u \in [0, 1]. \qquad (7)$$

where Δu is the Laplacian of u and $\operatorname{div}(d^k - b^k)$ is the divergence of $(d^k - b^k)$. Based on (7), u^{k+1} can be approximately obtained by a Gauss-Seidel iterative method [16]. After that, (5) is solved with respect to d. It has been shown in [19] that the minimizing solution d^{k+1} is given by soft-thresholding:

$$d^{k+1} = \frac{\nabla u^{k+1} + b^k}{|\nabla u^{k+1} + b^k|} \max(|\nabla u^{k+1} + b^k| - \mu^{-1} g_b, 0). \quad (8)$$

III. Our Method

In this section, we describe the proposed constrained active contour method, which extends the convex active contour model of (1) (originally designed for automatic image segmentation) for interactive image segmentation.

A. Contour Initialization

For any active contour method, the contour needs to be initialized before the contour evolution process. Here, we use the segmentation result of the Geodesic method [9] for contour initialization due to its fast processing speed and the ability to avoid the "small-cut" problem.

In particular, we represent the result of the Geodesic algorithm 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. In the Geodesic algorithm, for a pixel x, its geodesic distances to the foreground or background seed regions are computed, which are denoted as $D_F(x)$ and $D_B(x)$ respectively. Then, an estimate of the probability that the pixel x belongs to the foreground is calculated as

$$P(x) = \frac{D_B(x)}{D_F(x) + D_B(x)}.$$

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

It is worth mentioning that other region-based image segmentation algorithms such as Random Walks [6] can also be used for contour initialization. For example, the segmentation result of the Random Walk algorithm can be directly used as the probability map $P(x)$. The second column of Fig. 3 shows the probability maps of Geodesic and Random Walks.

B. Constrained Active Contour Model

As shown in (1), the convex active contour model consists of two terms: a regional term and a boundary term. Next, we discuss how to modify these two terms to incorporate the information from the user input and the initial segmentation result so as to ensure the refined contour complying with the user input.

1) Regional term formulation: The foreground and background seeds give an excellent description about the color distributions of the foreground and background regions. Foreground/background GMMs introduced in [20] 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 log likelihood that x belongs to foreground and background are $P_F(x) = \frac{-\log Pr(x|F)}{-\log Pr(x|F) - \log Pr(x|B)}$ and $P_B(x) = \frac{-\log Pr(x|B)}{-\log Pr(x|F) - \log Pr(x|B)}$. We incorporate this regional information derived from foreground/background strokes into the regional term of the convex active contour model as

$$h_r(x) = P_B(x) - P_F(x) \qquad (9)$$

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 $P_B(x) > P_F(x)$ (resp. $P_B(x) < P_F(x)$) and $P_B(x) - P_F(x)$ 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 (9) fails in the case that the foreground and background color models are not well separated. Thus, to avoid this problem and also to make use of the segmentation result obtained by the Geodesic algorithm in step 1, we further propose to incorporate the probability map $P(x)$ into the region term h_r as

$$h_r(x) = \alpha(P_B(x) - P_F(x)) + (1-\alpha)(1-2P(x)) \quad (10)$$

where α, $\alpha \in [0,1]$, is a tradeoff factor. The second term $(1-2P(x))$ in (10) 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, and vice versa.

It is important to properly set the tradeoff factor α in (10). When the foreground and background colors are well separable, it is desired that the first term in (10) becomes dominating;

otherwise, the second term in (10) should dominate. Thus, similar to the one suggested in [8], we set α to be the distance between the foreground and the background GMMs, which is defined as

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\log Pr(x_i|F) - \log Pr(x_i|B)}{\log Pr(x_i|F) + \log Pr(x_i|B)} \right| \qquad (11)$$

where n is the total number of pixels in the image.

In addition, it can be observed that when $h_r(x) \rightarrow +\infty$ (resp. $h_r(x) \rightarrow -\infty$), the regional term forces $u(x) = 0$ (resp. $u(x) = 1$) to minimize Eq. (1). This observation allows us to 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 ($P(x) > 0.9$ or $P(x) < 0.1$), 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 complies with the user input and also exploit more information from the initial segmentation result.

Note that unlike the h_r definition in section II-A, our proposed h_r model is fixed given the user input and the initial segmentation. Thus, there is no need for the three-step alternate iteration described in section II-A. Instead, only step 2 is needed, which can be solved by the Split Bregman method discussed in section II-B.

2) Boundary term formulation: The boundary term of $\int_{\Omega} g_b(x)|\nabla u|\, dx$ in (1) is essentially a weighed total variation of function u, where the weight g_b plays an important role. The definition of g_b in (2) is effective in the sense that it encourages the segmentation along the curves where the edge detection function is minimal. The problem with (2) is that at 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

$$g_b = \beta \cdot g_c + (1-\beta) \cdot g_e \qquad (12)$$

where g_c and g_e are the results of applying the edge detection to the GMM probability map $P_F(x)$ and the original image, respectively, and β, $\beta \in [0,1]$, is a tradeoff factor computed in a similar way as α given in (11). 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 compares the results with and without incorporating the edge detection of the GMM probability map. It can be seen that incorporating g_c enhances the conventional edge detection result g_e, especially at the weak edges, which leads to a more accurate boundary contour.

IV. EXPERIMENTAL RESULTS

A. Parameter Setting

Our proposed method has a few parameters, including $\{\lambda, T, \mu, \alpha, \beta\}$. The settings of α in (10) and β in (12) have

(a) Original image   (b) GMM map   (c) Result using (2)   (d) Result using (12)
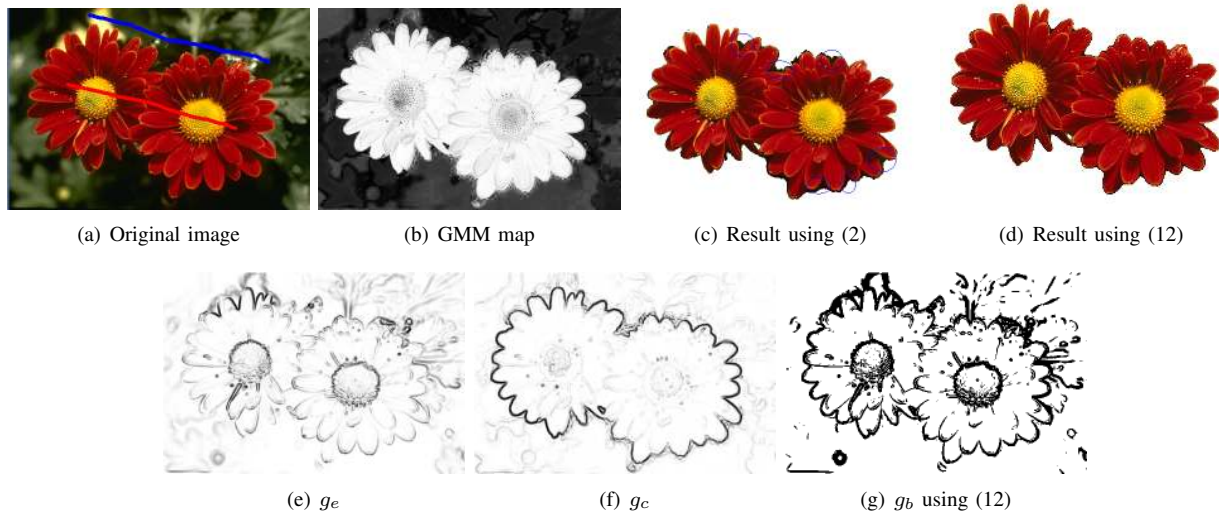
(e) g_e (f) g_c (g) g_b using (12)

Fig. 2. Comparisons of the results using the two different g_b definitions in (2) and (12), respectively. Note that some boundary problems due to using (2) are marked in (c).

been discussed in Sections III-B1 and III-B2, respectively. The parameter λ in (1) allows the user to control the scale and the smoothness of the segmentation. A smaller value of λ results in a smoother contour. We empirically set λ to be 100 in our experiments. The parameter T in (3) is set to 0.5, although the results are quite robust to different values of T. This is because the converged u values are typically close to either 0 or 1. The parameter μ in (5) is set to 10000 empirically. Note that all the parameters are set in the same way for all the experiments.

B. Test on The Benchmark Data Set

The commonly used MSRC ground truth data set [5] is chosen for testing and comparison. The MSRC data set contains 50 test images, each of which are provided with trimaps and ground truth. Table I summarizes the achieved error rates (percentage of mislabelled pixels) by different state-of-the-art interactive image segmentation algorithms and our proposed method. We also test several variants of our method, where we replace the Geodesic method by the Random Walks or the Yang's method [8]. For fair comparison, we use exactly the same trimaps provided by the MSRC data set as the user inputs for all the algorithms. The error rates for other state-of-the-art algorithms are either directly quoted from the best results reported in literature or obtained through our implementation. Note that the MSRC data set is somewhat biased because the provided trimaps only contain small unknown regions, for which Geodesic and Random Walks perform well. We still choose it since it is the only publicly available data set with trimaps provided.

From Table I, it can be seen that our proposed method achieves very low error rate, outperforming the state-of-the-art interactive image segmentation algorithms, including Geodesic [9], Random Walks [6], GrabCut [5] and Yang's approach [8]. In addition, the table suggests that our method is insensitive to the initial contour since the initializations using different methods lead to almost the same error rate (also illustrated in Fig. 3).

TABLE I
ERROR RATE COMPARISON USING THE MSRC DATASET WITH EXACTLY THE PROVIDED TRIMAPS.

Method	Error rate (%)
GMMRF [21]	7.9 (reported in [21])
Geodesic [9]	5.21 (our implementation)
Random Walks [6]	5.4 (reported in [22])
Segmentation by transduction [22]	5.4 (reported in [22])
GrabCut [5]	5.66 (reported in [8])
Yang *et al.* [8]	4.08 (reported in [8])
Our method	3.768
Our method with Random Walks	3.77
Our method with Yang *et al.*	3.765
Random Walks with AT [22]	3.3 (reported in [22])
Segmentation by transduction with AT [22]	3.3 (reported in [22])

We would like to point out that a lower error rate of 3.3% on MSRC data set has been reported in [22], where a segmentation by transduction (SBT) method was proposed. In fact, the SBT itself produces a high error rate of 5.4%, the same as Random Walks, for the MSRC dataset. When testing the MSRC dataset with the provided trimaps, the authors of [22] further proposed a simple adaptive threshold method (AT) as a post-processing step for Random Walks or SBT, which subsequently reduces the error rates of Random Walks and SBT from 5.4% to 3.3%, for the given trimaps. As explained in [22] itself, this large reduction in the error rate is due to the particular form of the seeds in this data set. As the unlabelled points only cover a small band along the object boundary, any naive segmentation approach such as AT that tracks the skeleton of the unlabelled points might perform quite well on this data set. There, the AT method is very specific to the MSRC data set or a boundary brush tool where the unseeded region only covers a small band along the object boundary. It does not work well for images with large and unregulated unknown regions, as shown in Fig 4.

In contrast, our approach is general and works well on images with large unknown regions. This is evident by all the figures shown in this paper, where the strokes are sparsely

(a) Input

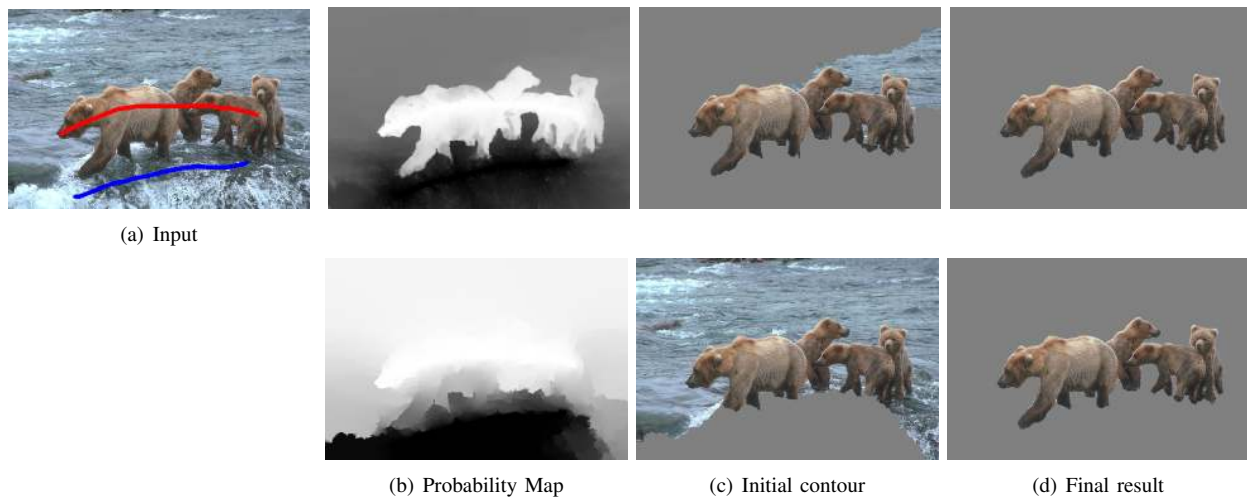(b) Probability Map   (c) Initial contour   (d) Final result

Fig. 3. The segmentation results of our methods initialized by either Geodesic [9] or Random Walks [6]. First row: our method (initialized by Geodesic); second row: our method initialized by Random Walks.

(a) Input image   (b) Trimap   (c) Result by RW with AT   (d) Our result

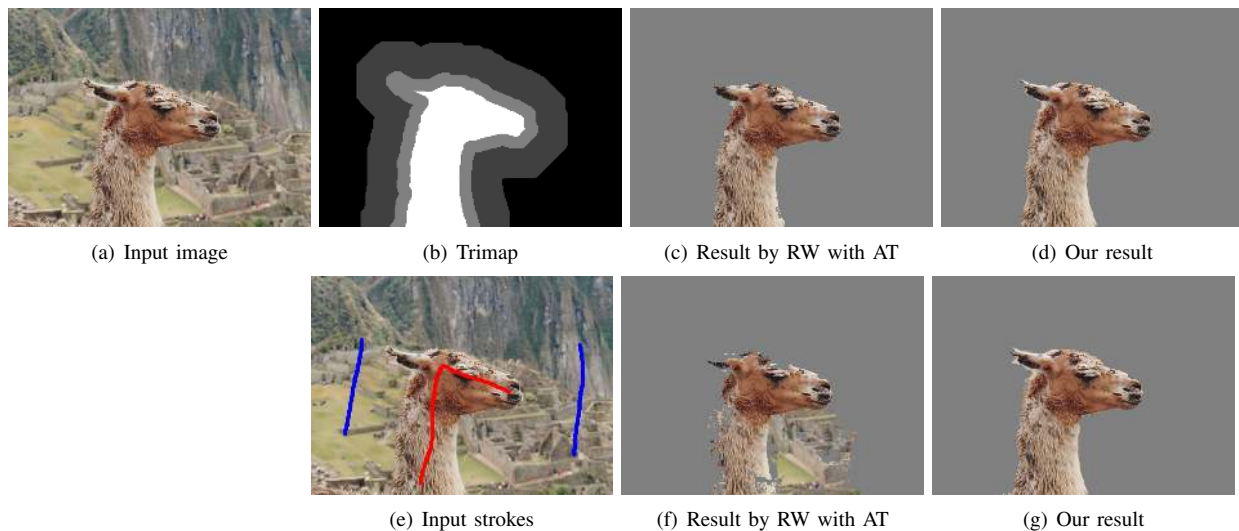(e) Input strokes   (f) Result by RW with AT   (g) Our result

Fig. 4. Comparison between our method and Random Walks (RW) with the adaptive thresholding (AT) method [22]. The RW with AT method works well with the provided trimap but fails with the input of sparse strokes.

drawn, leaving large unknown areas in the images. The good visual results (see Section IV-C) indicate that the performance of our method does not rely on a carefully designed trimap, as in the MSRC data set. Moreover, our segmentation tool implemented in C++ using the Geodesic method [9] for pre-segmentation runs very fast, taking less than three seconds in total to segment an image with a resolution of 640×480 on a PC with Intel 2.67 GHz CPU and 2 GB RAM, where the speed of convergence of our constrained active contour optimization costs less than half a second.

C. More Visual Results

Fig. 5 shows the segmentation results of different algorithms for three different images. It can be seen that in these cases with large unknown regions, Geodesic and Random Walks perform poorly, producing inaccurate and jaggy boundary contours. Although the performance of GrabCut is much better, its results still contain some clearly visible artifacts, e.g.

around the neck of the man, the right elbow of the boy, and the bottom of the boat. On the contrary, our method produces accurate and smooth contours that snap to geometry edges.

As we previously mentioned, the Random Walks and Geodesic algorithms are sensitive to the seed locations. Fig. 6 compares the segmentation results with different user inputs. It can be seen that, for Random Walks and Geodesic, different users inputs result in different segmentations. In contrast, our constrained active contour is able to fix the problem and generate stable results insensitive to the user input, as shown in Fig. 6.

Although the GrabCut algorithm is insensitive to the seed locations, it is sensitive to the number of seeds and has the "small cut" problem (see Fig. 1). Fig. 7 gives another example, where we want to cut out the fireman on the left. GrabCut tends to include the other fireman into the foreground region due to similar appearance. Even when more background strokes are drawn over the fireman on the right (Fig. 7(c)), Grabcut still chooses the minimum cut around the added strokes (Fig. 7(d)).

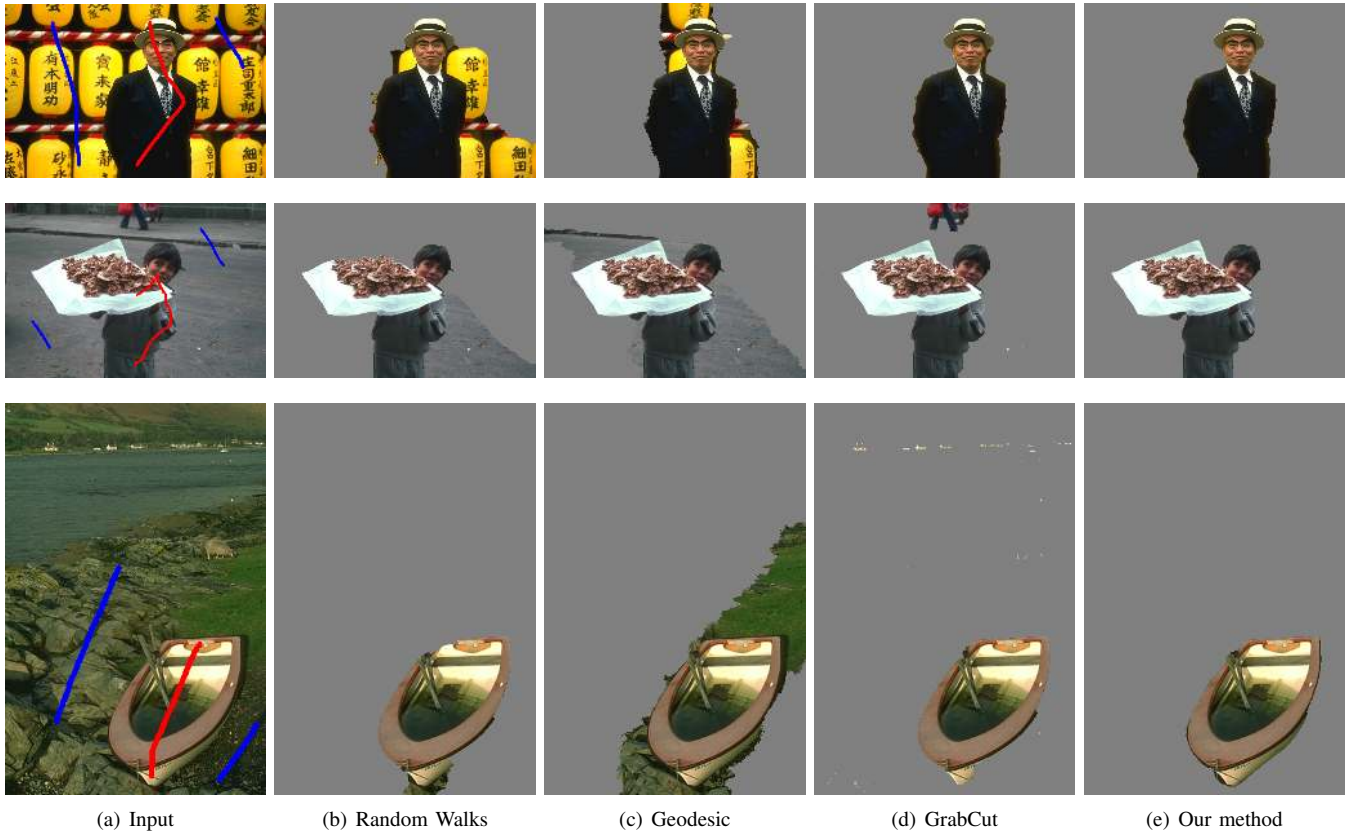(a) Input   (b) Random Walks   (c) Geodesic   (d) GrabCut   (e) Our method

Fig. 5. The segmentation results of different algorithms, including Random Walks [6], Geodesic [9] and GrabCut [5], and our proposed method, for three different images.

(a) Input   (b) Random Walks   (c) Geodesic   (d) Our method
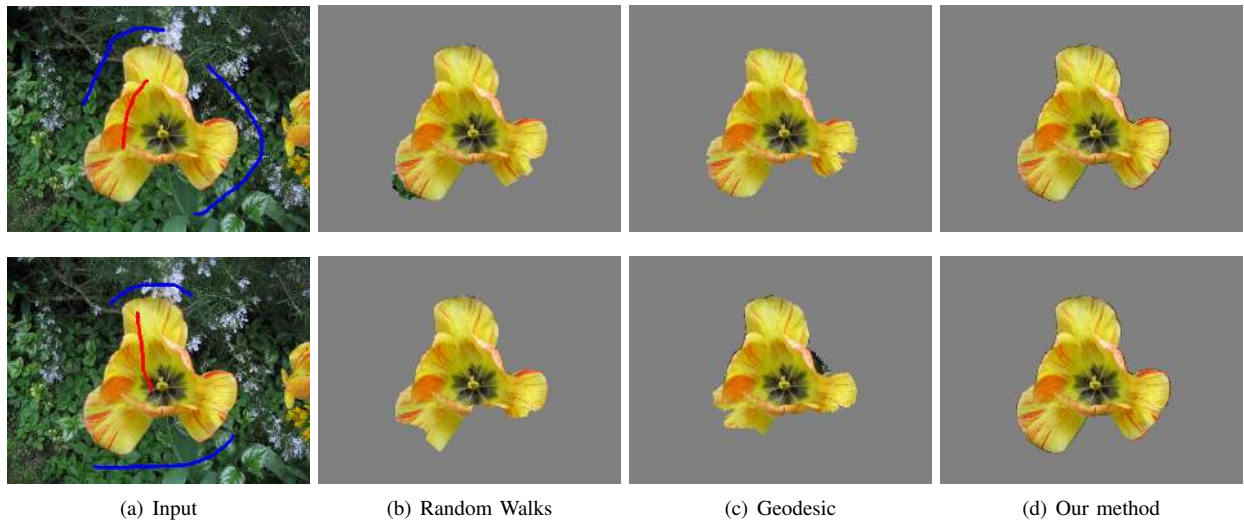
Fig. 6. The segmentation results with different user inputs.

Grabcut eventually produces the expected result after a large number of strokes. On the contrary, our method can produce the good segmentation result with only two input strokes. This is mainly because in our method the combination with the Geodesic / Random Walks helps propagate the seeds and introduces the spatial or location information into the active contour model, which significantly mitigates the small cut problem. Moreover, GrabCut heavily relies on the global color model, which often results in disjoint parts or noisy parts in the cases that the foreground and the background have similar color (see Figures 1, 5 and 7).

One important property of our constrained active contour model is the ability to handle topology changes of the boundary contour, which can not be achieved using the classical Snakes model. As shown in Fig. 8, while the Random Walks method produces two initial boundary contours that separate the object into two halves, our constrained active contour can evolve the boundary contours to one outer contour around the object and one inner contour at the object's left hand. Similarly, the geodesic method produces only one initial closed boundary contour while our method further produces the additional inner contour.

D. Limitations

The proposed method is essentially a hard segmentation method. It cannot handle transparent or semi-transparent boundaries such as semi-lucent hair. Another inherent limitation of the proposed method lies in its underlying assumption that the shape of the object is smooth and can be well described by the weighed shortest boundary length. It cannot handle very sophisticated shapes such as bush branches or hair. Fig. 9 gives two failure examples, where our method is unable to cut the deer antlers and the tree branches in a clean way.

V. Conclusions

In this paper, we have proposed a robust and accurate interactive image segmentation method based on the continuous-domain convex active contour model. We have demonstrated that our method outperforms the state-of-the-art interactive segmentation methods. It exhibits many desirable properties for a good segmentation tool, including the robustness to user inputs and different initializations, the ability to produce a smooth and accurate boundary contour, and the ability to handle topology changes. Our method runs very fast due to the fact that the proposed constrained active contour model can be solved quickly by a fast Split Bregman Method and the adoption of the Geodesic algorithm for initialization. We would like to point out that although the proposed constrained active contour model is able to automatically optimize an initial contour, it can also take additional user inputs for further user-guided contour evolving. This is especially necessary in either the case that the initial contour is very poor or the case that a highly accurate result is sought.

This work can be extended in a few ways. For example, it might be beneficial to apply the continuous-domain convex active contour model for other segmentation problems such as image matting or video segmentation. Also, it is interesting to adopt some advanced evaluation method such as the user simulation based approach proposed in [23] to fully evaluate the performance of different interactive image segmentation methods.

References

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *IJCV*, vol. 1, no. 4, pp. 321–331, 1988.

[2] E. Mortensen and W. Barrett, "Interactive segmentation with intelligent scissors," *Graphical Models and Image Processing*, vol. 60, no. 5, pp. 349–384, 1998.

[3] A. Falcao, J. Udupa, and F. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: live wire on the fly," *Medical Imaging, IEEE Transactions on*, vol. 19, no. 1, pp. 55–62, 2002.

[4] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in ND images," in *IEEE ICCV*, 2001, pp. 105–112.

[5] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH*, 2004.

[6] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[7] J. Zhang, J. Zheng, and J. Cai, "A diffusion approach to seeded image segmentation," in *IEEE CVPR*, 2010.

[8] W. Yang, J. Cai, J. Zheng, and J. Luo, "User-friendly interactive image segmentation through unified combinatorial user inputs," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2470–2479, Sept. 2010.

[9] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *IEEE ICCV*, 2007, pp. 1–8.

[10] A. Criminisi, T. Sharp, and A. Blake, "Geos: Geodesic image segmentation," *ECCV*, pp. 99–112, 2008.

[11] A. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *IEEE ICCV*, 2007, pp. 1–8.

[12] F. Chan, F. Lam, P. Poon, H. Zhu, and K. Chan, "Object boundary location by region and contour deformation," *Vision, Image and Signal Processing, IEE Proceedings*, vol. 143, no. 6, pp. 353–360, 2002.

[13] Bresson, X. *et al.*, "Fast global minimization of the active contour/snake model," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 2, pp. 151–167, 2007.

[14] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof, "Tvseg - interactive total variation based image segmentation," in *British Machine Vision Conference*, 2008.

[15] T. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM J. Appl. Math.*, vol. 66, no. 5, pp. 1632–1648, 2006.

[16] T. Goldstein, X. Bresson, and S. Osher, "Geometric applications of the split Bregman method: Segmentation and surface reconstruction," *Journal of Scientific Computing*, vol. 45, no. 1-3, pp. 272–293, 2010.

[17] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International journal of computer vision*, vol. 22, no. 1, pp. 61–79, 1997.

[18] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[19] Y. Wang, W. Yin, and Y. Zhang, "A fast algorithm for image deblurring with total variation regularization," *Rice University CAAM Technical Report TR07-10*, 2007.

[20] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis, "Improved fast Gauss transform and efficient kernel density estimation," in *IEEE ICCV*, 2003, pp. 664–671.

[21] Blake, A. *et al.*, "Interactive image segmentation using an adaptive GMMRF model," *ECCV*, pp. 428–441, 2004.

[22] Duchenne, O. *et al.*, "Segmentation by transduction," in *IEEE CVPR*, 2008, pp. 1–8.

[23] K. McGuinness and N. E. O'Connor, "Toward automated evaluation of interactive segmentation," *Computer Vision and Image Understanding*, vol. 115, no. 6, pp. 868–884, 2011.
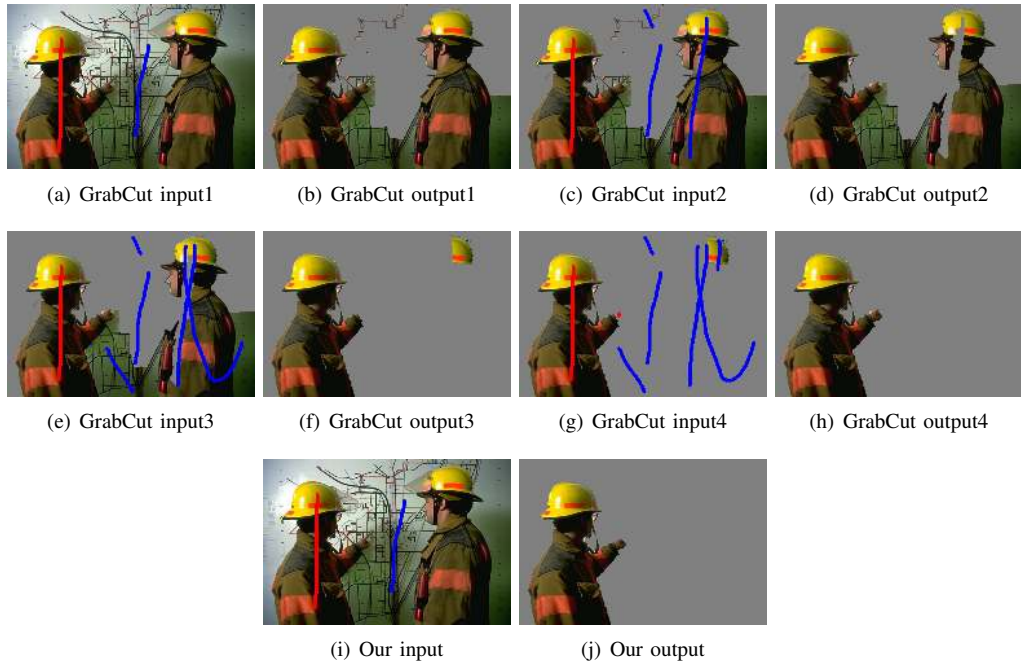
(a) GrabCut input1   (b) GrabCut output1   (c) GrabCut input2   (d) GrabCut output2

(e) GrabCut input3   (f) GrabCut output3   (g) GrabCut input4   (h) GrabCut output4

(i) Our input   (j) Our output

Fig. 7. Comparision bewteen GrabCut and our method. It can be seen that Grabcut [5] suffers the "small cut" problem while our method can quickly produce a good segmentation result.

(a) Input   (b) Random Walks   (c) Our method with Random Walks   (d) Geodesic   (e) Our method
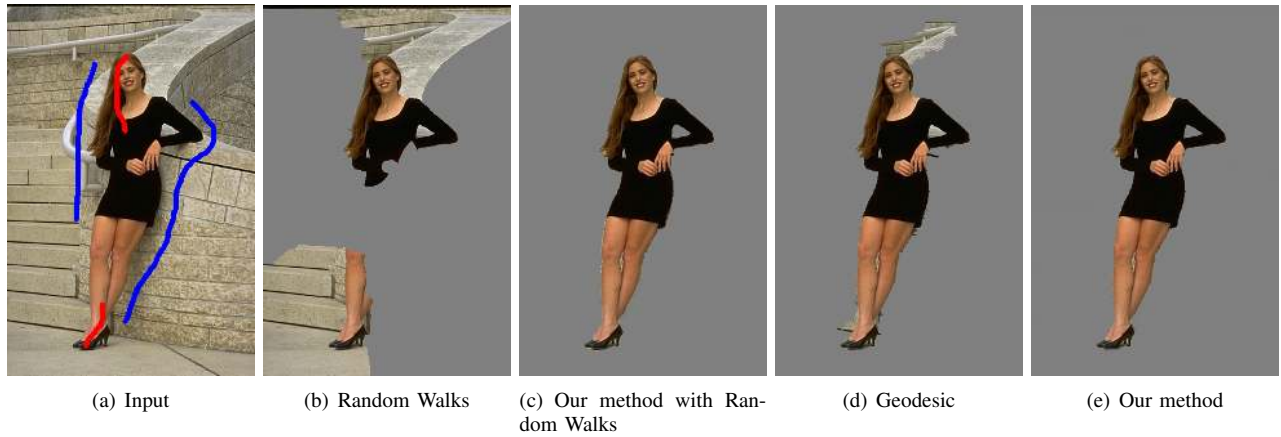
Fig. 8. An example to show the ability of our method on handling topology changes.
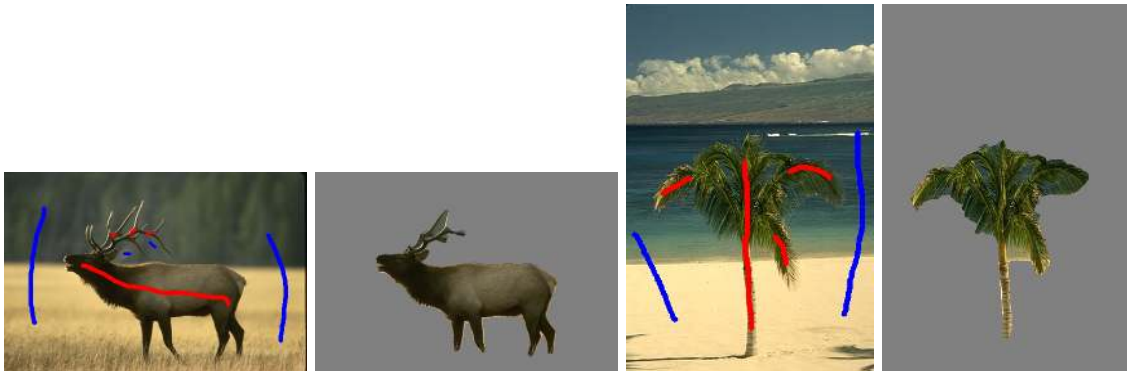

Fig. 9. Failure examples, where our method is unable to well handle very sophisticated shapes.

Thi Nhat Anh Nguyen received the B.Eng. degree in 2007, and is expected to receive the M.Eng degree in 2012, both from School of Computer Engineering, Nanyang Technological University, Singapore. She is currently a lecturer at the Information Technology Faculty, Danang University of Technology, Vietnam.

Juyong Zhang received the B.S. degree in computer science and engineering from the University of Science and Technology of China, and the Ph.D. degree in computer engineering from NTU (Nanyang Technological University), Singapore, in 2011. From June 2010 to July 2010, he was a visiting student in Massachusetts Institute of Technology. From 2011, he worked as an exchanging Ph.D student and then a Postdoctoral Research Fellow in computer science of EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland. His research interests include computer graphics, geometry processing, image processing, architecture geometry, as well as numerical PDE.

Jianfei Cai (S'98-M'02-SM'07) received his PhD degree from the University of Missouri-Columbia. Currently, he is an Associate Professor and also the Head of Visual & Interactive Computing Division at the School of Computer Engineering, Nanyang Technological University, Singapore. His major research interests include visual signal processing and multimedia networking. He has published more than 100 technical papers in international conferences and journals. He has been actively participating in program committees of various conferences. He serves as the leading Technical Program Co-Chair for IEEE International Conference on Multimedia & Expo (ICME) 2012 and the leading General Co-Chair for Pacific-rim Conference on Multimedia (PCM) 2012. He was an invited speaker for the first IEEE Signal Processing Society Summer School on 3D and high definition / high contrast video process systems in 2011. He is also an Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT), and a senior member of IEEE.

Jianmin Zheng received the BS and PhD degrees from Zhejiang University, China. He is an associate professor in the School of Computer Engineering at Nanyang Technological University. His research interest includes computer aided geometric design, computer graphics, CAD, animation, visualization and interactive digital media. He has published more than 100 technical papers in international conferences and journals. He was a technical papers committee member for the inaugural SIGGRAPH Asia 2008. He is an associate editor of The Visual Computer.