INTERACTIVE SEGMENTATION USING CURVE EVOLUTION AND RELEVANCE FEEDBACK

M. A. El Saban and B. S. Manjunath

Electrical and Computer Engineering Department, University of California, Santa Barbara
{msaban, manj}@ece.ucsb.edu

Abstract

We propose in this paper an interactive segmentation algorithm based on curve evolution techniques. The task of automated segmentation has proven to be highly complex and application dependent. User's knowledge can be used to alleviate the problem. In this paper, we propose the use of a recently developed curve evolution technique [1], augmented with a relevance feedback phase through user interaction. After the initial automatic segmentation is computed, the user presents his positive/negative feedback via a simple user interface. Segmentation parameters are then adapted locally to reflect user's requirements. Experimental results show the usefulness of the proposed approach in interactive segmentation tasks.

1. INTRODUCTION

Image segmentation has been the subject of numerous research works in the past years, as it is an important preliminary step in many vision applications. Fully automatic techniques do not require manual effort, yet suffer from the fact that segmentation is an an ill posed problem. Depending on the application at hand and on the user, different segmentation results can be sought. On the other hand, interactive segmentation approaches require manual effort but can render the problem more well posed. In this context, semi-automated methods with little user's interaction are a potentially attractive alternative in applications such as interactive graphics and animation. The goal is to produce a final acceptable segmentation with minimal and, perhaps, imprecise user input. Our belief is that semi-automated/ interactive methods provide a good tradeoff for the image segmentation problem.

In an interactive segmentation scheme, the system exploits the knowledge presented by the user to aid the segmentation process. An initial segmentation result is computed and presented to the user. The user provides his feedback which will then be used to modify the segmentation. The requirements in implementing such a feedback system include: a) fast response time, b) user friendly interface, c) taking care of user feedback imprecision, and d) dealing with complex objects in cluttered backgrounds. Most previous interactive segmentation schemes enabled user interaction in an input stage by means of marking some pixels as objects or drawing a initial rough segmentation curve. In this paper, we propose the novel idea of enabling user's feedback on a pre-computed segmentation as a post-processing stage.

In a typical segmentation, there are three main possible sources of segmentation "errors" (or unsatisfactory results) that can occur in a segmentation output (as discussed in [2]): a) an edge is missing due to weak image edges, b) an edge is produced at a shifted position due to the presence of a nearby stronger edge and c) an edge is present which the user is not interested in (this is the perhaps, easiest case as it can be fixed using region merging at the end).

We propose an interactive segmentation scheme for tackling the segmentation problems mentioned above. We note from the above types of errors that the main source of the first two errors is a weakness in the edge function strength at the location of interest with potentially higher edge strength values in a nearby location (as in the second error). These two errors are tackled via the feedback mechanism. Starting with a fully automated segmentation technique, we modify its methodology in computing the segmentation result by incorporating user's feedback.

The rest of the paper is organized as follows. Section 2 reviews the related work. In section 3, we give an overview on the used curve evolution technique in our segmentation approach. The user feedback module is introduced in section 4. Section 5 gives experimental results of the proposed scheme as well as some discussion.

2. RELATED WORK

The first approaches to semi-automated segmentation are based on changing a set of parameters that affect the final

This research was supported in part by the following grants and awards: ONR # N00014-01-0391 and NSF # ITR-0331697

segmentation, such as adjusting a threshold. These methods are heuristic, and in most cases, the user is not aware of the direct effect of parameters on the final result, thus the desired segmentation result is rarely achieved. The next step in the evolution of semi-automated segmentation was in the use of snakes and curve evolution-based techniques as in [3]. Similar ideas are used with watershed-based methods [4],[5]. These methods improved the results significantly, but can lead to missing edges where the computed image edge functions are weak. Attempts have been made to overcome leaking problems by imposing constraints on boundary length, but these may miss true irregular object boundaries in some cases.

The second group of related work tackles the image segmentation problem within a knowledge-guided scheme. Example work include [6] where the authors propose the use of graphical shape models to represent the prior knowledge about objects within a statistical scheme. Similarly, in [7], human 3-D shape model is used to confirm hypotheses about human presence in scenes, thus guiding their segmentation from the scene. In medical image analysis knowledge-based segmentation has gained great attention as in [8] because of the difficulty of using a fully automated general-purpose segmentation method. Moreover, many of the objects of interest have ill-defined boundaries, such as a tumors [9]. Knowledge-based segmentation is also applied to other domains such as SAR images [10] and document images [11].

The different works we discussed here require an *offline* training phase to learn shape or object models, hence cannot adapt to every image and user's need. Furthermore, these methods are very specific to the application at hand, hence hard to generalize to other domains. For example, the work in [12] specializes in segmenting left-ventricles, so it cannot be applied to MRI images easily.

The last semi-automated image segmentation research direction we discuss here is characterized by more user interaction in the segmentation task with the use of improved real-time drawing and interfacing techniques to speed the segmentation process [13],[14],[15], [16]. These are mainly based on a fusion of image processing and graphics techniques. The most prominent work in this direction is the intelligent scissors approach [13] with its different variants [17]. It provides a user-friendly interface to trace object boundaries with simple computer mouse clicks. However, its methodology in tracing the boundaries can make it tedious to trace boundaries of complex textured objects in cluttered background. In addition, it still relies on image features, namely gradients that may be weak in some interesting parts of the image from a user's point of view.

3. CURVE EVOLUTION SEGMENTATION

Region-based segmentation is a very basic area of segmentation, but curve evolution techniques that are region-based are only recently being developed [18, 19]. We are using in this work the automatic segmentation technique presented in [1]. This segmentation algorithm is based on a competition of region-based forces. The objective of this segmentation is to increase similarity inside each region and simultaneously increase dissimilarity across regions. This criterion is expressed as a cost function from which the following curve evolution equation is derived:

$$\frac{\partial C}{\partial t} = \left(\alpha \int\limits_{R_o} w(c,s)ds - \beta \int\limits_{R_i} w(c,s)ds \right) \vec{N} \quad (1)$$

where R_i is the foreground, R_o the background as in a two-region cases. $w(c,s)$ is the similarity measure between two pixels, and c is a point on the curve C. The choice of the similarity measure $w(c,s)$ directly effects the segmentation. In this work, we define w from feature vectors F computed at each pixel in the image: $w(c,s) = \sum_{i=1}^{N} |F_i(c) - F_i(s)|$, where $F_i(s)$ is the i^{th} component of the feature vector F at pixel location s.

The region-based formulation discussed above can be used to compute a segmentation between the object and the background. However, using a region-based approach only can result in inaccurate boundaries, hence the region-based part is complemented by an edge-based part. For this purpose, an edge vector field (EVF) is utilized EVF is designed in such a way that its vectors point towards the closest discontinuity. The design of EVF is inspired by the design of Edgeflow vector field [20].

To calculate the direction of EVF vectors, at each pixel we look for the highest probable edge direction. Assume that σ is the scale at which we are looking for edges. Let I be the intensity image - extensions to multi-valued images is straightforward. Let $\hat{I}_\sigma$ is the Gaussian smoothed image at σ. At pixel $s(x,y)$ and orientation θ, the prediction error $Er(\sigma,\theta)$ is defined as:

$$Er(\sigma,\theta) = \left| \hat{I}_\sigma(x+4\sigma\cos\theta, y+4\sigma\sin\theta) - \hat{I}_\sigma(x,y) \right| \quad (2)$$

The larger the error, the higher is the possibility of having an edge at the direction θ. Instead of finding the direction with the largest error value which may be prone to noise, we define edge probability at direction θ as:

$$P(\sigma,\theta) = \frac{Er(\sigma,\theta)}{Er(\sigma,\theta) + Er(\sigma,\theta+\pi)} \quad (3)$$

Using probabilities at each direction, the edge direction at

$s(x, y)$ is calculated as:

$$\arg\max_{\theta} \int_{\theta-\pi/4}^{\theta+\pi/4} P(\sigma, \theta')d\theta' \quad (4)$$

Using this direction, the edge vectors $\vec{S}(\sigma)$ at a specific scale σ are calculated as the vector sum:

$$\vec{S}(\sigma) = \int_{\theta-\pi/4}^{\theta+\pi/4} [\ Er(\sigma, \theta')\cos(\theta') \quad Er(\sigma, \theta')\sin(\theta')\ ]d\theta'$$

To calculate this vector field for multi-valued images, such as texture and color images, we only need to change the error calculation.

To create a curve evolution using EVF, a cost function is defined that encourages the curve to pass through the discontinuities while minimizing the length of the curve, leading to the following curve evolution:

$$\frac{\partial C}{\partial t} = V\kappa\vec{N} - (\vec{S}\cdot\vec{N})\vec{N} \quad (5)$$

where V is the edge function computed as an approximate inverse gradient of EVF.

By integrating (5) with (1), we can update the curve evolution equation as the following:

$$\frac{\partial C}{\partial t} = \underbrace{L(R_0, R_i)}_{\text{Region}-\text{Based}} + \underbrace{V\kappa\vec{N} - (\vec{S}\cdot\vec{N})\vec{N}}_{\text{EVFbased}} \quad (6)$$

where $L(R_0, R_i)$ is the region-based term as defined in (1).

4. RELEVANCE FEEDBACK USING FEATURE SELECTION AND CONTROLLED CURVE SPEED

Based on the segmentation technique we discussed in the previous section, the segmentation result is computed in an automated fashion. As we noted earlier, in most cases the result could have some unsatisfactory parts from the user's point of view. If we allow the users to present their positive / negative *feedback* to the system, the segmentation result can be improved significantly. The feedback process has two main issues to be addressed: a) how the feedback is presented?, and b)how the feedback information is used?

We propose to use a user friendly interface to enable the user to present his feedback. The user simply marks pixel blocks –of controllable size– as being positive or negative feedback regions. We use the term *positive* to note an area where the user is interested to create an edge that is missing. Negative feedback, on the other hand, means that the user wants to remove an already computed edge.

Depending on the type of feedback entered by the user, we pursue different steps in modifying the segmentation result. First, consider the positive feedback case. We argue here that in most of the interesting cases, from an analysis point of view, an edge would be missed because of a weak local image gradient. Other cases where a user's interesting edge is missing due to absence of gradient can be dealt with using visual drawing interfaces or using object models. Here, we only consider the case of a missing edge due to a weak gradient, and possibly a nearby higher gradient values that attracts the segmentation boundary.

We start with the observation that when dealing with multi-valued images -such as color images - we can compute different gradient magnitudes if we consider each of the image channels separately. For example, an image with shades of red color will have no blue gradient components, while having possibly high gradient magnitudes based on the red channel. In the automatic segmentation part, we consider a single global weighting of different image features when computing the error in (2). Based on the user's feedback, the weighting can be adjusted locally in a more efficient manner to capture the user's preference. This is similar in spirit to the relevance feedback mechanism used in information retrieval tasks. We propose a *feature selection* strategy to enhance the gradient at positive feedback areas marked by the user. In this way, we choose the feature that increases the error in (2) at areas of positive feedback. Our algorithm is as follows:

- Compute the error in (2) for each of the N channels (RGB color and Gabor texture features) of the feature vector F at areas of positive feedback.
- Use a double argument maximization on both $theta$ and each of the features i for $1 \leq i \leq N$ in the feature vector F:

$$\arg\max_{\theta,F} \int_{\theta-\pi/4}^{\theta+\pi/4} P(\sigma, \theta', F_i)d\theta' \quad (7)$$

- By solving the maximization in (7), we compute the best EVF direction and feature that will result in the highest gradient.

Next, we consider the case when the user marks a group of pixels by negative feedback. In such a case, the user is interested in skipping the computed edge and may want to create an edge in the vicinity. The most straightforward way of ignoring the edges at negative feedback areas is to increase the curve speed at these areas such that the segmentation curve doesn't get trapped at these locations. For the areas marked with negative feedback, we use a constant high speed in the curve evolution equation instead of the curve evolution equation in (6). It is worth to note that for the unmarked pixels (neither positive or negative), the automatic segmentation algorithm discussed in the previous section is applied without changes.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

We present some preliminary experimental results of our proposed interactive segmentation system using both positive and negative feedback. First, we show in Fig.1.a-c the use of positive feedback, where the user is interested in the missed edge at the person's shoulder. Fig.1.d-i show examples of negative feedback where the user wants to get rid of edges inside the face and tiger body. Initial curves are started at desired parts of the image and are allowed to evolve according to (6) using a level set implementation. When all curves converge, the final segmentation is obtained. Using the feedback algorithm detailed in section 4, the positive and negative feedback are implemented. Positive feedback is supplied from the user by drawing an arc at the missing edge position. Whereas the negative feedback is marked by mouse clicks on unwanted edges.

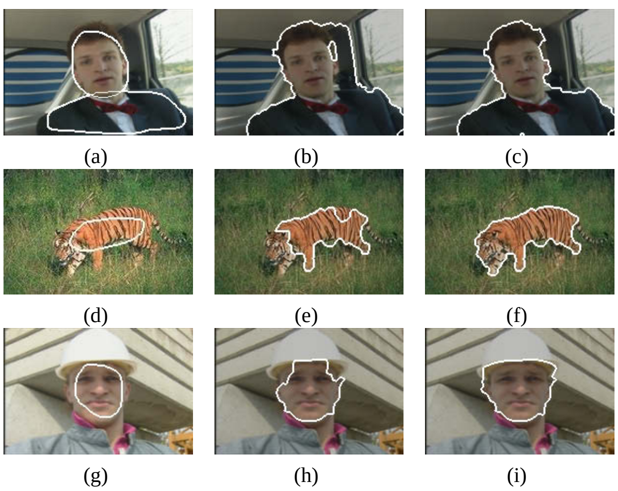

Fig. 1. User's feedback illustration using a color image (best seen in the pdf version). In each row a different image is shown with the initial curve shown on the left image, the result of initial (automatic) segmentation in the middle and finally the segmentation result after considering user's feedback is shown at the right.

5.1. Discussions

We proposed in this work an interactive segmentation algorithm based on curve evolution techniques. Since automatic segmentation of generic images is an ill posed problem, an amount of user interaction can render the problem more tractable. In this paper, we proposed the use of a recently developed region-based curve evolution technique. After the initial segmentation result is computed, the user presents his positive/negative feedback via a simple user interface. Segmentation parameters are then adapted locally to reflect user's requirements. Experimental results are promising. Currently, we are looking into other techniques to incorporate user's feedback such as locally adaptive scale selection.

6. REFERENCES

[1] B. Sumengen et al., "Image segmentation using multi-region stability and edge strength," in *ICIP 2003*.

[2] T. Deschamps, *Curve and Shape Extraction with Minimal Path and Level-Sets techniques. Applications to 3D Medical Imaging*, Ph.D. thesis, Paris-Dauphine university, 2001.

[3] B. Sumengen et al., "Image segmentation using curve evolution and flow fields," in *ICIP 2002*.

[4] F. Zanoguera et al., "A toolbox for interactive segmentation based on nested partitions," in *ICIP 1999*.

[5] H. Nguyen et al., "Watersnakes: energy-driven watershed segmentation," *TPAMI 2003*.

[6] A. Neumann, "Graphical gaussian shape models and their application to image segmentation," *TPAMI 2003*.

[7] T. Zhao et al., "Segmentation and tracking of multiple humans in complex situations," in *CVPR 2001*.

[8] A. Amini et al., "Flexible shapes for segmentation and tracking of cardiovascular data," in *ICIP 1998*.

[9] M. Clark et al., "Automatic tumor segmentation using knowledge-based techniques," *IEEE Transactions on Medical Imaging 1998*.

[10] S. Haker et al., "Knowledge-based segmentation of sar data with learned priors," *TIP 2000*.

[11] K. Lee et al., "Geometric structure analysis of document images: a knowledge-based approach," *TPAMI 2000*.

[12] S. Lei et al., "A knowledge-based boundary delineation system for contrast ventriculograms," *IEEE Tran. on Inf. Tech. in Biomedicine 2001*.

[13] E. Mortensen and W. Barrett, "Intelligent scissors for image composition," in *Proceedings of conf. on Comp. graph. '95*.

[14] O. Gerard et al., "Real-time interactive path extraction with on-the-fly adaptation of the external forces," in *ECCV 2002*.

[15] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV 2001*.

[16] K. Tan and N. Ahuja, "Selecting objects with freehand sketches," in *ICCV 2001*.

[17] W. Barrett and E. Mortensen, "Toboggan-based intelligent scissors with a four-parameter edge model," in *CVPR 1999*.

[18] A. Tsai et al., "A statistical approach to snakes for bimodal and trimodal imagery," in *ICCV 1999*.

[19] T. F. Chan and L. A. Vese, "Active contours without edges," *TIP 2001*.

[20] W. Ma and B. Manjunath, "Edgeflow: A technique for boundary detection and image segmentation," *TIP 2000*.