This paper appears in: IEEE International Conference on Systems, Man and Cybernetics, 1999

Color Image Segmentation Editor Based on the Integration of Edge-Linking, Region Labeling and Deformable Model

Jean Gao, Akio Kosaka and Krisnawan Rahardja
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana 47907
jgao@ecn.purdue.edu, kosaka@ecn.purdue.edu
rahardja@ecn.purdue.edu

Abstract

A new general-purpose color image segmentation editor (CIE) for the purpose of extracting semantic object is designed, implemented and tested on a number of various natural scene images. Our system integrates deformable models and image statistics including intensity, color, gradient and texture. First, a low-complexity edge-linking algorithm is proposed as a post-processor to Canny operator, which basically builds regions of smooth intensity surfaces by closing all dangling edges. The topologically-based region labeling makes full use of relationship among pixels and produces good label image. To refine the extracted object of interest (OOI), the deformable model based on energy minimization is applied. Experimental results suggest the efficiency and accuracy of the system in segmentation operation.

1 Introduction

The advanced development of digital camera during the past few years rapidly changed the concept of camera industry, since users tend to use computer-based photo storage and retrieval rather than traditionally film-based photo collection. Such users want to edit and process photo images based on their interest by using computer terminals; therefore, the camera industry tends to move to software-related domain so that users can entertain new contents of documents and photographs.

In such domains, the extraction of semantic object of interest (OOI) becomes significantly important, which is also a necessary unit for the future intelligent cameras capable of object-based capture, manipulation and storage. Except commercial applications, image editing scheme offers a vital tool for three dimensional medical image reconstruction, prework for image database search, etc.. So far, only a few commercial products, such as *Photoshop* from *Adobe*, *Image Magick*, and the newly released *Ultimatte KnockOut*, are available for segmenting color images. Unfortunately, they generally don't take full advantage of users' preference or require too much interaction from users. More specifically, the computer algorithms either don't know exactly what regions should be extracted or don't produce satisfactory result for images with complicated backgrounds.

An ideal image editor allows an image to be edited according to user's preference. One of the most common task consumer interested in is to segment out the semantically meaningful object. It is an easy task for human vision system to locate the visually coherent object, but it is hard for the computer to understand which part of the image the user wants. So a completely automatic algorithm is impossible at least in current computer domain [4]. But the computer may perform other editing operations more efficient than the human being, such as line and circle drawing.

In this paper, we propose a new system concept that can handle such difficulties. Our system emphasizes the human-interactions to capture the user's interests through the user-friendly interface. Different from other interactive methods, our region-based editing system greatly minimizes the user's interaction; its memorizing and reversible scheme offers the user friendly error-correction capability. It also fully takes into account of color information so that a more precise and robust segmentation can be achieved under a busy background. In the following sections, we will demonstrate the strategies of our color image segmentation editor as well as the experimental results.

0-7803-5731-0/99/$10.00 ©1999 IEEE

2 Initial Segmentation and Edge Linking

Our hybrid approach starts with an edge-linking algorithm as post-processor to Canny [2] operator. It is known that Canny operator produces dangling edges when edge tracker can not find the next neighboring pixel with non-maximum suppressed strength greater than given low threshold value. Since typically this threshold is set to the lowest possible value, desired solution must be independent of this thresholding procedure. Our analysis indicates it is plausible to choose the next pixel based on 1) the closest characteristic at the the dangling edge, and 2) smoothly continuing the edge fragment.

As a solution to 1), a pixel characteristic is qualified by a vector of size M whose entries are its edge strengths in M directions(i.e. 15°, 30°, 45°, etc.) in addition to Canny's standard horizontal and vertical directions. Note that this operation do not add much computational complexity from original Canny operator since they can be done simultaneously with the standard edge strengths. Then the candidate pixel that maximizes the dot-product of its characteristic vector and that of its corresponding dangling edges should be chosen as the next edge pixel.

For constraint 2), as illustrated in Fig.1, candidate pixels for a dangling edge (center 'x') are the three pixels (bullets) that are the farthest from the next edge pixel (square) on the fragment, so the momentum of the fragment is sustained. As a pre-processing, the edge image has to be thinned [9] to provide edge width of one pixel. To speed up the process, linking is done iteratively where in each iteration all dangling edges are grown by one pixel, as linking of two proper dangling edges in opposite directions can be done in half number of iterations. The process terminates when any of the candidates is an edge pixel.

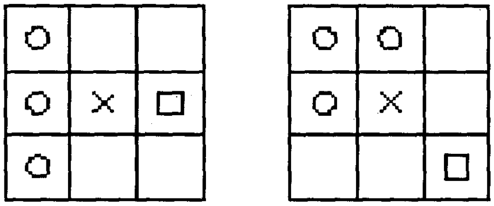


Figure 1: Candidates of a dangling edge. Bullets are the candidates, 'x' is the dangling edge and square is the edge pixel.

In a word, edge fragment can be closed by repeatedly adding into its membership one pixel from neighboring candidates that maximizes the dot-product until an edge from another different edge-fragment is found. Simplicity and low computational complexity are its main strengths, and the results are comparable to any other popular methods. Fig. 2(b) demonstrates the initial segmentation result after edge-linking with vector size M of 4 and with region colors are shown as the corresponding mean value.

3 Interactive Region Editing

The editing function allows the user to add and remove any regions preferred, and cut or add partial part of any region.

3.1 "Select" Operation

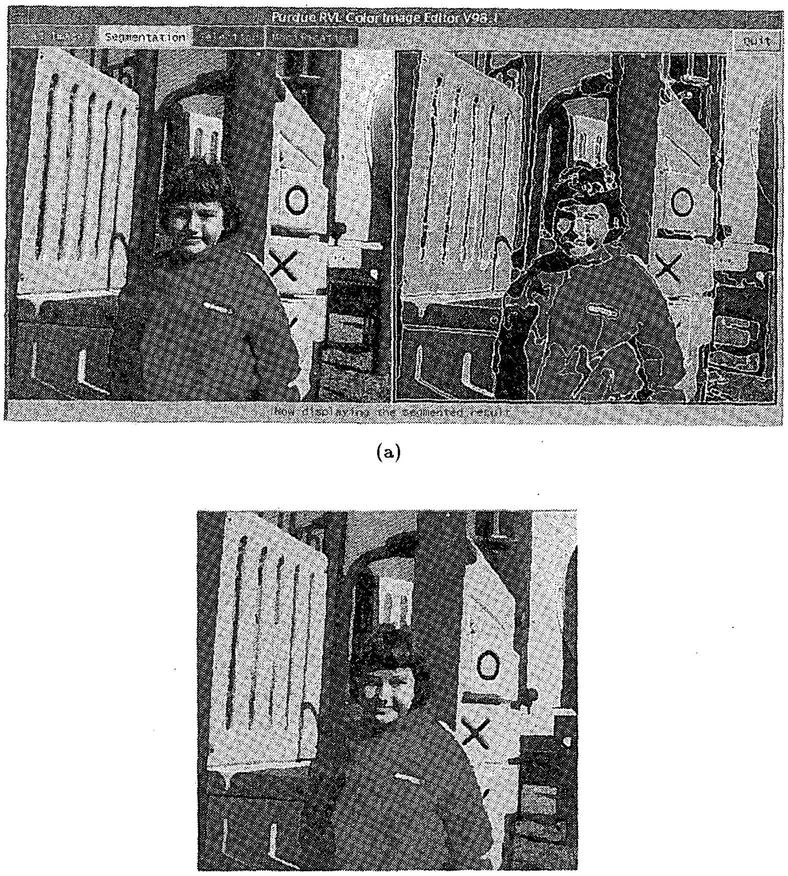
"Select"-"Start"-"Accept": Based on the image obtained by combination of modified edge-map and the mean value regions, this operation allows the user to judge the single region easily. Instead of tracing the boundary manually or grouping the possible edges [3], the user can quickly determines the region candidates by simplify clicking single once on those regions intended to compose the visually coherent object. To further reduce human interaction, it is only required the selected regions form the enclosure of the object of interest(OOI) regardless of the business and the size of the inside of the enclosure.

"Select"-"Eliminate": If the user misselects some regions not wanted, this operation offers the region removing function. The above two functions are independently and can be operated repeatedly, which provides the user great flexibility.

"Select"-"Fill"-"Done": Once the enclosure is determined, all selected regions are given label "1" as object region; and the background and holes are given label "0". Next reversing labeling is done, that is, the single object is labeled as "0", background is labeled as "1", and the hole are labeled as "2", "3".... For the regions except region "0" whose cables are NOT "1", they are holes and are filled by their original values. Fig. 3 explains the process. Fig. 4 (a) shows the region selection by using mouse, (b) demonstrate the result after the hole-filling operation.

3.2 "Modify" Operation

During the initial segmentation, when part of background has almost same feature as the object of inter-



Now displaying the segmented result

(a)

(b)

Figure 2: An example of color image editor (CIE). (a) CIE interface. (b) Segmentation after edge-linking and region-labeling.

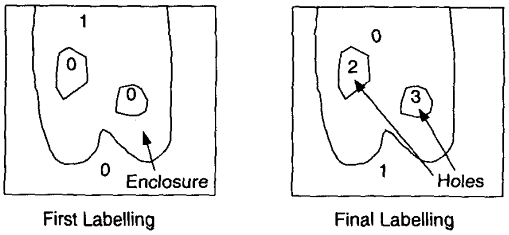


Figure 3: Hole-filling of the enclosure.

est, it is usually misclassified as the background. The *Modify* operation gives the user the ability to make corrections in case this occurs though not always necessary.

To do the modification, the user initiates by clicking the region wanting to keep, then traces the rough boundary by mouse. The 4-component region-labeling is utilized to renumber the region candidates composing the semantic object. The labeling process applies features of adjacency and connectedness. After the *Modify*, the new traced points become edge points in the edge map, and the cut-off region becomes the new region in the background.

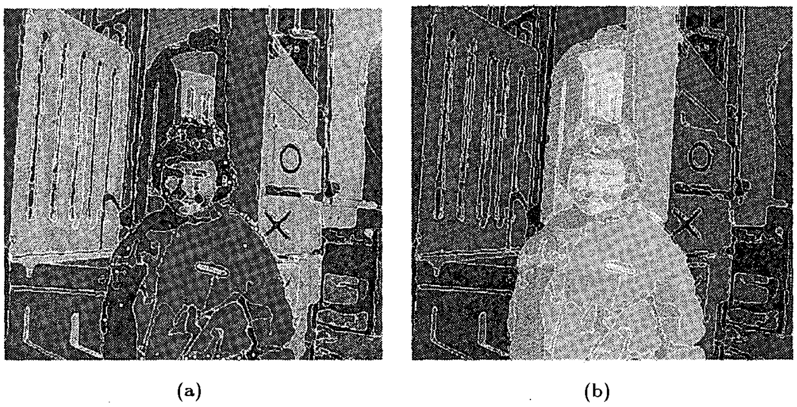


Figure 4: Interactive Editing (a) Effective region selection. (b) User preferred regions after hole filling.

4 Boundary Refining

Variational calculus based energy minimization contour modeling (also called Snake) since its first being introduced by Kass *et al.* [6] has received much attention and been improved significantly. Our interest in this approach is motivated by the need for refining the object of interest (OOI) especially when accuracy becomes the focus.

A snake is an energy-minimizing continuous spline whose energy is guided by the image forces that attract it to the wanted image features and the internal energy that serves as a smoothness constraint. The contours slither while always minimizing the energy function, and therefore act like snakes and exhibit dynamic property.

The total energy of a spline with parametric description of $S(t) = (x(t), y(t))$ can be described as

$$\begin{aligned} E^*_{snake} &= \int_{snake} E(S(t))\, dt \\ &= \int_{snake} [E_{int}(S(t)) + E_{img}(S(t)) \\ &\quad + E_{con}(S(t))]\, dt \\ &= \int_{snake} [E_{int}(S(t)) + E_{ext}(S(t))]\, dt. \qquad (1) \end{aligned}$$

Before implementing the above snake algorithm, a recursive algorithm is used to non-uniformly sample the control points [5] based on curvature and distance. The basic criterion is the sharper the curvature, the more control points are allocated.

Dynamic programming is used here to minimize the global energy function in Eq. (1) [1]. It is a discrete multistage decision process that varies the control points at each iteration in the direction of a global energy minimum. During each iteration, the optimal energy functional is obtained from all possible shifts of the control points to new positions within an $m \times m$ neighborhood of the current placement of the control points.

5 Discussion

This paper has presented a user-in-the-loop semantic object segmentation scheme. Region-based unit selection greatly reduces the human interaction. With the ability to add and remove regions, it is powerful for complicated image. The *Modify* function offers the user error correction ability if applicable. It can be implemented on any kind of image modality without special training stage. The robustness of above system has been tested on various kinds of color images. Fig.2, 4, 5 show an example of some of above procedures. Our Color Image Editor system (CIE), though still in its prototype stage, provides a quicker, less human interaction, more efficient, more reproducible, and more accurate general purpose object extraction tool than other existing human-interactive image editing systems. Not only for consumer-based commercial use, it can also be applied to medical image volume rendering, pre process for image database search and query.

More future work can be done on automatic weight parameters estimation during the energy-

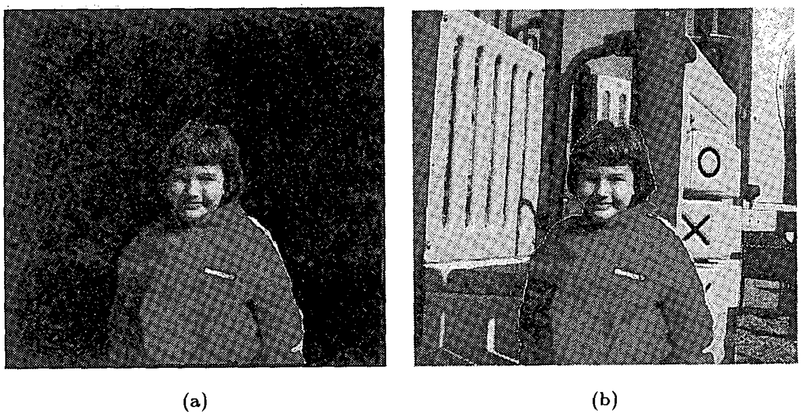


Figure 5: (a) Region modification. (b)Refined contour imposed on original image.

minimization refining stage. Possible directions also include the compression of the extracted object, and extending the domain to image sequences.

References

[1] A. Amini, T. Weynouth, and C. Ramesh, "Using Dynamic Programming for Solving Variational Problems in Vision," *IEEE Trans. Pattern Anal. Machine Intell*, vol. 12, pp. 855-867, 1990.

[2] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741, Nov. 1984.

[3] J. H. Elder and R. M. Goldberg, "Image Editing in the Contour Domain," *Proceedings of International Conference on Pattern Recognition,* Santa Clara, California, 374-381, 1998.

[4] A. X. Falcao, J.K. Udupa, S. Samarasekera, and S. Sharma, "User-Steered Image Segmentation Paradigms:Live Wire and Live Lane," *Graphical Models and Image Processing,* 60, 233-260, 1998.

[5] J. Gao, A. Kosaka, and A. Kak, "A Deformable Model for Human Organ Extraction," *Proceedings of International Conference on Image Processing,* Chicago, Illinois, 1998.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Mo dels," *Int J. Comput. Vision*, vol. 1, no. 4. pp. 321-331, 1988.

[7] Y. G. Leclerc and S. W. Zucker, "The Local Structure of Image Discontinuities in One Dimension," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 3, pp. 341-355, 1987.

[8] K. Rahardja and A. Kosaka, "Vision-Based Bin-Picking: Recognition and Localization of Multiple Complex Objects Using Simple Visual Cue," *Proceedings of IEEE/RSJ Intelligent Robots and Systems Conference,* 1996.

[9] A. Rosenfeld and A. Kak, *Digital Picture Processing*, 1982, Academic Press.