

cated background. In the following sections, we will present the details of our color image segmentation editor including experimental results.

2 Initial Region-Based Segmentation Using Edge-Linking

Our hybrid approach starts with a coarse region segmentation which applies the Canny's operator [?] to an original image, and then performs an edge-linking algorithm as the post-processor to the Canny's operator. It is known that the Canny's operator produces dangling edges when edge tracker cannot 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 the original Canny's 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 'o') 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 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.

In summary, edge fragments 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

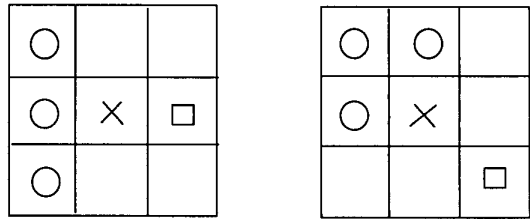


Figure 1: *Candidates for connecting a dangling edge. Circles represents the candidates, 'X' shows the dangling edge, and square is the edge pixel.*

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 parts of any region.

3.1 "Select" Operation

"Select"-"Start"-"Accept": Based on the image obtained by combination of the 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 determine the region candidates by simply clicking once on those regions intended to compose the visually coherent object. To further reduce human interactions, it is only required that the selected regions form an enclosure of the object of interest (O^2I) regardless of the complexity and the size of the inside of the enclosure.

"Select"-"Eliminate": If the user misselects some regions undesired, this operation offers the function of removing regions. The above two functions are independently and can be operated repeatedly, which provides the user with a great flexibility.

"Select"-"Fill"-"Done": Once the enclosure is determined, all selected regions are given label "1" as an object region; and the background and holes are given label "0". Next a reversing labeling is done, that is, the single object is labeled as "0", background is labeled as "1", and the holes 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 the mouse, and Fig. 4 (b) demonstrates the result after the hole-filling operation.



(a)



(b)

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

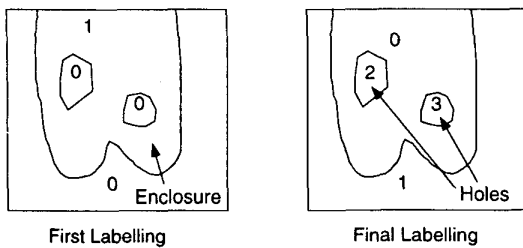


Figure 3: Hole-filling of the enclosure.

3.2 “Modify” Operation

During the initial segmentation, when part of background has almost the same feature as the object of interest, it is usually misclassified as the object of interest. The *Modify* operation gives the user with the ability to make corrections in case this occurs though not always necessary.

To do the modification, the user initiates by clicking the region to be kept, 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.

Each of the above two functions, “*Select*” and “*Modify*”, can be executed multiple times until satisfactory results are obtained or no further progress is achieved.

4 Boundary Refining

Variational calculus-based energy minimization contour modeling (also called *Snake*), since its first being introduced by Kass *et al.*, has received much attention and been improved significantly. Our interest



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

in this approach is motivated by the need for refining the object of interest (O^2I) especially when the accuracy becomes focus.

The active contour model aims to the contour extraction by minimizing the sum of (1) the internal energy given by the connectivity and smoothness constraints and (2) the external energy to represent the fidelity to the image features. The contours slither while always minimizing the energy function, and therefore act like snakes and exhibit dynamic properties.

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

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

Before implementing the above snake algorithm, a recursive algorithm is used to non-uniformly sample the spline to obtain most representful control points based on curvature and distance [5]. Briefly speaking, its basic criterion is that the sharper the spline, 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.

4.1 Color-Feature Based Edge-Force Definition

Color images provide more ample information to the computer than gray-scale images, but they also bring more possible choices and uncertainty for optimal solutions. To attract the contour of the desired boundary, normally the gradient of the image intensity is used as the edge force, i.e., $E_{edge} = -|\nabla I|^2$. For the color image, it is found that shading, shadow and highlights will also induce the change of E_{edge} which is not desired at the boundary area. To prevent the contour from being misattracted, the psychophysical variable saturation in HSI space is used as part of the external force. Due to its numerical instability at low saturation, the E_{edge} based on the intensities of R, G, B channels is applied at low saturation. Briefly speaking,

$$\begin{aligned}
 &\text{when } Saturation > Threshold, \\
 &\quad E_{edge} = -|\nabla Saturation|^2 \\
 &\text{else} \\
 &\quad E_{edge} = -1/3(|\nabla R|^2 + |\nabla G|^2 + |\nabla B|^2)
 \end{aligned}$$

4.2 Region Energy

To provide a better representation for an optimal boundary, the region energy is defined as

$$E_{Region_k} = I(x, y)R_k \|\Delta_k I(x, y)\|^2 \quad (2)$$

where $I(x, y)$ is the image function, R_k is the relevant region and $\|\Delta_k I(x, y)\|^2$ is the squared error for region R_k during certain iteration. Region energy for unions of disjoint regions is defined as the combination of individual region and their corresponding Ward distance [7].

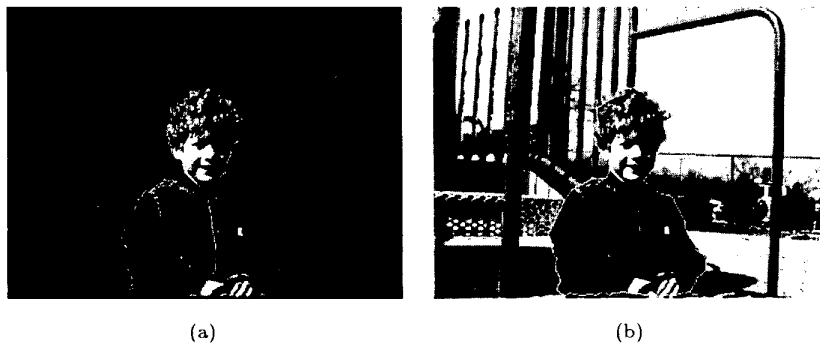


Figure 5: (a) Region modification. (b) Refined contour imposed on original image.

5 Concluding Remarks

The goal of this paper is to create a *user-in-the-loop* segmentation scheme of a semantic object which can process efficiently and more precisely. The region-based selection greatly reduces the human interaction. With the ability to add and remove regions, it is powerful for complicated images. The *Modify* function offers the user error correction ability if applicable. It can be applied to any types of image modalities without special training stages. The robustness of the above system has been tested on various kinds of color images. Fig.2, 4, 5 show an example of some of above procedures. We believe that our Color Image Segmentation Editor (CISE), 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 the automatic parameter estimation during the energy-minimization refining stage. Possible directions also include the compression of the extracted object, and extending the method to image sequences by considering both the spatial and temporal features.

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 Models," *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.