ROBUST INTERACTIVE IMAGE SEGMENTATION WITH AUTOMATIC BOUNDARY REFINEMENT

Dingding Liu [†] [‡] *Yingen Xiong* [†] *Linda Shapiro* [‡] *Kari Pulli* [†]

[†] Nokia Research Center, Palo Alto, CA 94304, USA
[‡] Department of Electrical Engineering, University of Washington, USA

ABSTRACT

We propose an effective image segmentation approach with a novel automatic boundary refinement procedure that requires little user interaction and makes the object cutout process more robust and convenient. It achieves these goals by the following three steps. First, merge over-segmented regions according to the maximal similarity rule using a few marking strokes as input. Second, detect possible erroneous low-contrast object boundaries by analyzing image content. Third, automatically refine those boundary regions using both local and global information. Experimental results are good even on very complex images.

Index Terms— Interactive image segmentation, object cutout, graph cut, mobile image editing, mobile image processing

1. INTRODUCTION

Interactive image segmentation has many applications in image processing, computer vision, and computer graphics. In many image editing tasks, the goal is to cut out an object from its background. Small amount of user input is desirable on any device, but especially on mobile devices with small screens and inaccurate input methods, it is crucial that only few inputs are required. It is even more attractive when the exact boundary is detected automatically, since zooming in and adding strokes or dragging markers so they precisely align with a complex object boundary is tedious both on desktop and mobile devices. However, all interactive image segmentation methods we are aware of require some further input from users to refine boundaries when the initial algorithm fails. Generally, the user has to put markers or scribbles along the boundary, or even has to do the segmentation all over. Our work is the first attempt to automatically detect a possibly erroneous low-contrast object boundary and refine it by optimization using local and global information.

Though there has been much recent interest in interactive image segmentation [3, 4, 5, 6, 7, 8], our work is mostly related to [3] and [4]. The main steps in Li et al. [4] are object marking followed by boundary editing. Later research by Ning et al. [3], confirmed also by our studies, reveal that the graph-cut optimization on over-segmented regions does not work well when the set of input strokes is sparse. In other words, enough samples of foreground and background are needed. For example, if the foreground and background both have red color and only foreground red regions are marked by the user input, all red areas will be classified as foreground in the result. The small neighborhood of the penalty term that only considers directly adjacent regions may also cause problems.

The source images in Fig. 2 are taken from the Berkeley Segmentation Dataset and Benchmark[1], the website of [2] and PASCAL Visual Object Classes Challenge 2009 (VOC2009) dataset, except the forth row.

In [3] the color histogram of a region is used as the feature to perform maximally similar region merging (MSRM) to merge small regions into the background. The RGB colors are uniformly quantized into 16 levels per channel and histograms of $16 \times 16 \times 16 = 4096$ bins are used to estimate region similarity. This resulted in better segmentation quality than if graph cuts were used with the same strokes. However, the limitation of this approach is that only local information is considered in the merging process. It may fail when the user's markers do not cover all the main features of the object and background. It may also lead to the background annexing part of the foreground that is slightly more similar to the adjacent background region than to the immediate adjacent foreground regions. Moreover, calculating a histogram with 4096 bins in each over-segmented region is time-consuming and not suitable for mobile phones.

Aiming to make the interactive image segmentation more robust and to require as little user effort as possible, we propose the following novel algorithm consisting of three major steps. First, over-segmented regions are merged according to the MSRM rule with the input of a few strokes. This is inspired by [3], but instead of the RGB histograms, the mean color of each region in CIELab space is used as the feature. Second, suspicious low-contrast object boundaries are detected by adaptively thresholding the boundary regions. Third, suspicious boundary regions are relabeled by incorporating local information of pixels and global information of regions.

Our main contributions are that we (i) propose a robust approach for image segmentation with automatic boundary refinement; (ii) detect suspicious low-contrast object boundaries automatically by analyzing image contents; (iii) refine the suspicious low-contrast object boundaries by optimizations using both local and global information without user interaction; (iv) incorporate more efficient features into the energy function to make the approach more robust and avoid expensive feature calculation; (v) compare with other approaches to demonstrate the advantages of ours; and (vi) implement the algorithm on mobile phones.

We start by briefly introducing the work flow (Section 2) and then give more details of our segmentation and automatic boundary refinement (Section 3). We describe experiments and analyze results (Section 4) and finally summarize our conclusions (Section 5).

2. SUMMARY OF THE METHOD

Figure 1 shows the work flow of our approach. The strokes are first input to extract sampling of foreground and background of the source image. After over-segmenting the source image to generate many regions, they are merged into background and foreground using the MSRM rule [3], producing the initial image segmentation. Next, suspicious low-contrast object boundaries are detected. Pixels in those boundary regions are re-classified to decide which class the boundary region belongs to and the region is re-labeled if neces-

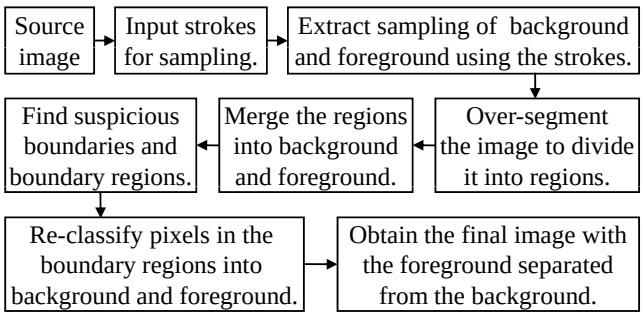


Fig. 1. Work flow of our method.

sary. After all suspicious boundary regions are processed, the final segmented image is obtained.

3. DETAILS OF THE APPROACH

We first describe our approach to initial over-segmentation and region merging, then the detection of suspicious low-contrast boundary, and finally the boundary refinement.

3.1. Merge initially segmented regions

Similar to [3] and [4], the image is first over-segmented. Since the mean-shift algorithm [9] preserves the boundaries well, it is used for the initial over-segmentation. After the user marks the foreground and background regions with short strokes, the background regions are merged using the MSRM rule [3]. However, instead of using computationally expensive color histograms, the mean color of each region is used. Then the initial labeling of each region, either foreground (marked as 1) or background (marked as 0), is generated. Our initial over-segmentation outputs regions that are bigger than 20 pixels. We also assume that the strokes provided by the user are sparse but pick the visually distinct regions. Given these two conditions, one round of boundary refinements was enough in our experiments.

3.2. Detect suspicious low-contrast object boundary regions

We want to find candidate regions for more careful analysis. We only need to consider regions at the boundary between the foreground and the background, and we only need to consider the ones whose colors are similar but whose labels differ.

The boundary regions U_{bd} are defined as the regions that have at least one neighboring region with a different initial labeling. For example, if a foreground region A_i has a neighbor B_j that is marked as background, then A_i and B_j are boundary regions. We also require that such regions share a boundary that is at least 4 pixels long.

For each boundary region, the mean color μ_{bd} is calculated, and the neighbor of opposite label with the most similar mean color is found. Finally we sort all the boundary regions U_{bd} according to their minimal color differences $d_{bd}^{i,j}$, and select the regions with $d_{bd}^{i,j} < d_{Thresh}$ as the suspicious low-contrast object boundary regions to be refined. The threshold d_{Thresh} is simply the median color difference over all boundary region pairs such that one region is foreground and the other one is background.

3.3. Refine suspicious boundary regions

After all suspicious low-contrast object boundary regions are detected, they are analyzed and possibly reclassified. Here we assume that the initial segmentation using the mean-shift algorithm includes the correct region boundary. Using the local and global information of the pixels inside each region, each pixel is classified to be foreground or background. Then the number of foreground and background pixels are counted inside each region. If one region has more foreground pixels, it is classified as a foreground region, otherwise as background.

We formulate this as a binary labeling problem of all the pixels belonging to the suspicious low-contrast object boundary regions, with different energy terms than previous work [4]. Our algorithm not only considers the pixels' similarity to their neighbors, but also their similarity to the regions marked by the user in terms of color means and standard deviations.

Suppose all the pixels in the suspicious low-contrast object boundary regions form a graph $G = \langle \upsilon, \varepsilon \rangle$, where υ is the set of all pixels and ε is the set of all arcs connecting the four adjacent pixels. The algorithm assigns a unique label x_i for each pixel $i \in \upsilon$, where $x_i = 1$ if i belongs to foreground and $x_i = 0$ if i belongs to background. The energy function we try to minimize is

$$E(X) = \sum_{i \in \upsilon} E_1(x_i) + \lambda \sum_{(i,j) \in \varepsilon} E_2(x_i, x_j), \tag{1}$$

where E_1 is likelihood energy, E_2 is prior energy, and $\lambda = 1$ in our experiments.

E_1 encodes the color similarity of a pixel to the marked foreground or background, taking into account the color standard deviation (std) within the regions, which can be regarded as the simplest texture information. They are also used to weight the color difference and std difference.

For each pixel i, suppose $C(i)$ is its color, μ_n^F is the mean color of marked foreground regions, μ_m^B is the mean color of marked background regions, and $\sigma(i)$ is the std of the region which it belongs to. σ_n^F is the foreground region std, and σ_m^B is the background region std. The following distances are computed:

$$\begin{cases} dm_i^F &= min_n \parallel C(i) - \mu_n^F \parallel \\ dm_i^B &= min_m \parallel C(i) - \mu_m^B \parallel \\ d\sigma_i^F &= min_n \parallel \sigma(i) - \sigma_n^F \parallel \\ d\sigma_i^B &= min_m \parallel \sigma(i) - \sigma_m^B \parallel \end{cases}. \tag{2}$$

Then, $E_1(x_i)$ is defined as follows:

$$E_1(x_i = 1) = \frac{z^F}{z^F + z^B} \tag{3}$$

$$E_1(x_i = 0) = \frac{z^B}{z^F + z^B}, \tag{4}$$

where $z^X = \frac{1}{\sigma(i)\sigma_n^X} dm_i^X + \sigma(i)\sigma_n^X d\sigma_i^X$ for $X \in \{F, B\}$.

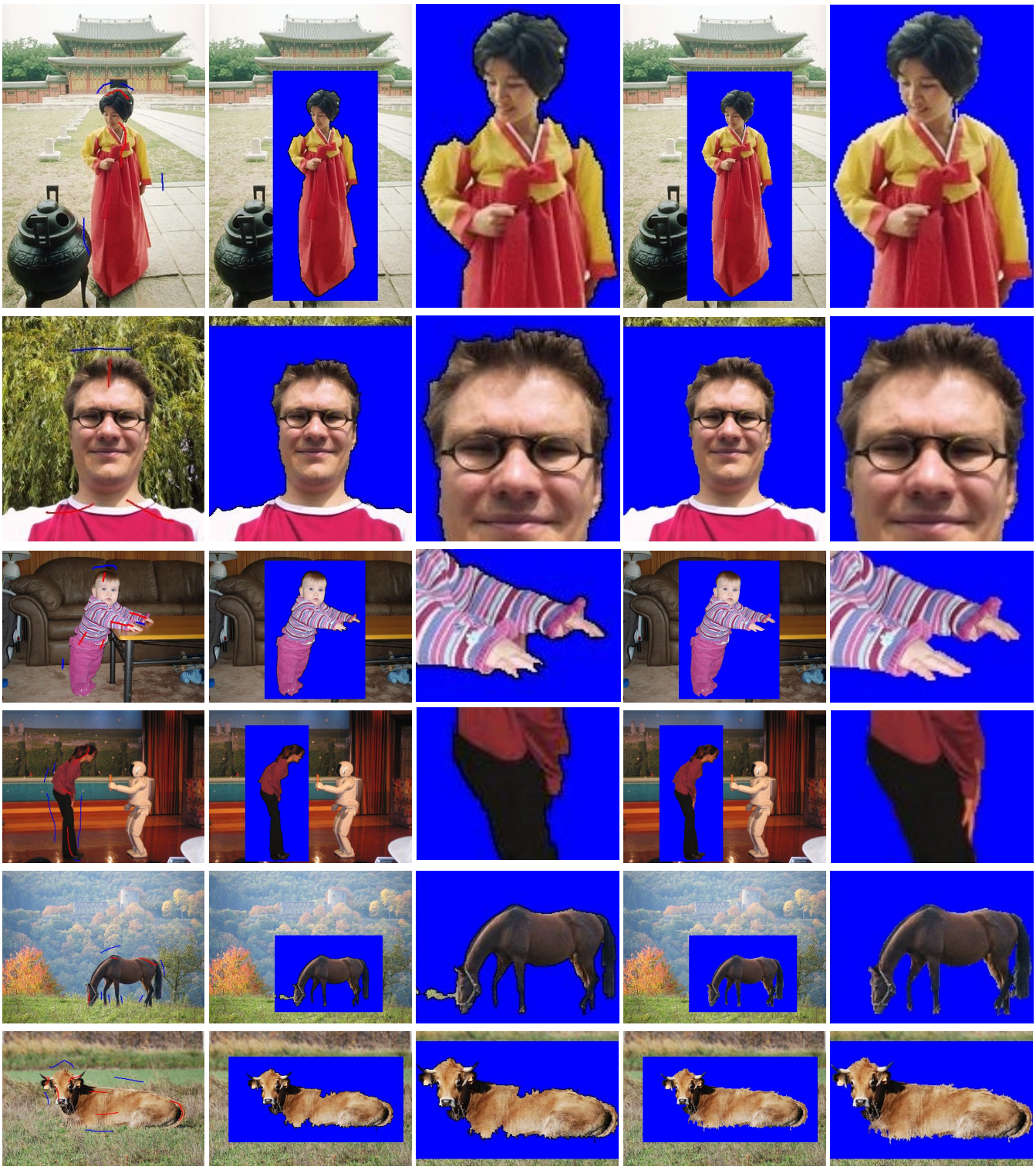
Because the marked boundary regions have been eliminated from Section 3.2, the Likelihood energy is different from that of equation (2) in [4]. The std of regions is also included, and the intuition is that if the region has high color std, d_{σ_i} is more important in the comparison, otherwise d_{m_i} is more important.

Similar to [4], E_2 is the energy due to the gradient along the object boundary. It also acts like a smoothing term, enforcing that similar neighboring pixels have the same label.

$$E_2(x_i, x_j) = \frac{|x_i - x_j|}{(\|C(i) - C(j)\|^2 + 1) \times scale}. \tag{5}$$

The difference is that the energy between pixels belonging to different regions is scaled down, so that the cut through the region boundary is facilitated. In our experiment, $scale = 1$ if pixel i and j belong to the same region, $scale = 2$ otherwise.

To minimize the energy function, we use the max-flow library [10]. Note that the assumption that the initial over-segmentation includes all correct boundary segments is crucial for good results.



(a) (b) (c) (d) (e)

Fig. 2. Comparisons between the method in [3] and the proposed algorithm. (a) The source images with user input stokes, where blue strokes are used to mark the background regions, and red strokes are used to mark the foreground regions. (b) Results obtained by the method in [3]. (c) Enlarged boundary region of the results obtained by method in [3]. (d) Results obtained by the proposed algorithm. (e) Enlarged corrected boundary region by the proposed algorithm.

4. APPLICATIONS AND RESULT ANALYSIS

Here we present some results and comparisons to [3] using region mean color to demonstrate the advantages of our approach. Our purpose in each application is to cut out the object that the user indicates with red strokes. Table 1 provides quantitative comparisons, and shows that our algorithm labels more pixels correctly.

Table 1. Comparison of correctly labeled pixel percentages.

Percentage (%)	Woman	Man	Baby	Asimo	Horse	Cow
Our algorithm	98.62	98.53	98.95	98.23	97.45	97.34
MSRM	96.96	97.73	97.89	97.66	95.56	94.77

Visual comparison in Fig. 2 also shows the better results provided by our approach. The first row shows a comparison of the proposed algorithm and [3] applied to a woman's image. Since the dress is mostly yellow and red, an inexperienced user may think marking those two is enough (a). However, due to the reflection on the shoulder, those regions are merged with the background (c) when only local neighboring information is used. Our approach can automatically detect those regions and correctly classify them (e) without further user input.

The second row shows a comparison of the algorithms to an image with a man in front of a highly textured background. With only three strokes marking the foreground and one simple stroke marking the background, [3] can segment the person for the most part, but fails at the ear on the left and upper hair part as (c) shows. The close-up image (e) shows that our algorithm can automatically refine those two places and construct a better segmentation. We also want to emphasize that our approach is different from matting the boundary. This image has been used in many previous studies. For example, our algorithm requires much less user input than Fig. 4 in [5]. We do not model the foreground and background using Gaussian Mixture Models (GMM) like [2], as use of GMMs may create problems [9]. Moreover, in the boundary refinement part, our algorithm does not need to perform the entire iterative minimization.

The third row compares the algorithms on a baby image. The challenge in this image is that the color of the baby's fingers is similar to that of the table. (a) shows the strokes which the user gave for marking the foreground (the baby) and the background. From (c) we can see that the fingers are not selected correctly and part of the table is labeled as foreground. With the proposed algorithm those regions were detected automatically and relabeled correctly. The baby is completely cut out.

The image in the fourth row poses a similar challenge when the user is trying to select the woman, but the input strokes miss her hands. Again, the hand colors are similar to the background, and they are omitted when only local neighborhood information is used (c). On the other hand, our algorithm incorporates global information, and the hands are relabeled as foreground in the refinement stage since those parts are similar to the woman's marked face color.

The fifth row shows our algorithm's robustness when the user does not provide enough background strokes. In (c), a part of the background grass became foreground as it is locally more similar to the gray strings on the horse head. Our algorithm correctly labels grass as background (e).

The sixth row shows our algorithm's robustness when the user's input strokes are sparse. The cow is very difficult to segment correctly as it has very similar color to the background (c), but our algorithm can again produce a better segmentation (e).

Through all the comparisons we can see that given the same set of user input strokes, our method with boundary refinement can segment the objects better. It is robust to the user inputs.

5. CONCLUSIONS AND FURTHER WORK

Interactive image segmentation is a challenging problem, and even more challenging on mobile phones as memory and processing power is more limited, and accurate marking is more difficult than on desktop. We have presented an effective robust interactive image segmentation algorithm with a novel automatic boundary refinement procedure that requires less user input and is robust to the quantity and locations of strokes. Both local and global information is considered in the boundary refinement process. The most obvious advantage of our approach is that it can save the user efforts in making the boundary better. Compared to zooming in and manually adding extra strokes or markers to correct the erroneous boundary segments, it is more convenient and more friendly when applied on mobile phones. The algorithm has been implemented on a Nokia N900 phone with an ARM Cortex-A8 600 MHz processor, 256 MB RAM, and a 3.5 inch touch-sensitive widescreen display. We plan to further improve the speed of the proposed algorithm and combine it with other image editing operations, to facilitate more elaborate, yet easy image editing.

6. REFERENCES

[1] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE Proc. Int. Conf. Computer Vision*, 2001.

[2] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH 2004 Papers*. ACM, 2004, p. 314.

[3] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445–456, 2010.

[4] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, 2004.

[5] X. Bai and G. Sapiro, "Distance cut: interactive segmentation and matting of images and videos," in *IEEE Intl. Conf. on Image Processing (ICIP)*, 2007, vol. 2, pp. 249–252.

[6] Y. Boykov and M.P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in ND images," in *International Conference on Computer Vision*. Citeseer, 2001, vol. 1, pp. 105–112.

[7] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive GMMRF model," *Lecture Notes in Computer Science*, pp. 428–441, 2004.

[8] E.N. Mortensen and W.A. Barrett, "Intelligent scissors for image composition," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM New York, NY, USA, 1995, pp. 191–198.

[9] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 603–619, 2002.

[10] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1124–1137, 2004.