Automatic Extraction of Object Region from Photographs

Takeshi Saitoh, Kimiya Aoki, and Toyohisa Kaneko

Toyohashi University of Technology,
1-1 Hibarigaoka, Tempaku-cho, Toyohashi, 441-8580, Japan
{saitoh, kaneko}@vcl.ics.tut.ac.jp

Abstract. This paper presents a new method for automatically extracting an object region from a photograph based upon a well-known method "Intelligent Scissors" (IS). For our application, it will be shown that (1) the cost should not be based on accumulated cost adopted by IS but rather on average cost and (2) only a few past pixels are needed for deciding the future route. It will be shown that our new method will be able to extract object region with a correct rate of approximately 93% if object images are well-focused.

1 Introduction

There are a number of objects of interest in our environment such as flowers, trees, birds, dogs, cats, cars etc. Sometimes they interest us so much to urge us to know their name of species. Digital cameras are becoming handy so that pictures can be taken easily, invoking recognition of object species. In such a case, a taken picture inevitably includes surrounding background in addition to an object of interest. The first task of recognition is to extract an object of interest from the background.

We have been attempting to recognize a natural flower[1] from a photograph; we used a black sheet on which we placed flowers picked from their natural habitat. To avoid this picking which is undesirable from the conservation of nature viewpoint, we propose in this paper an automatic method for extracting an object region from a photograph of a flower in bloom.

There are a number of methods proposed for the problem of extracting an object region from two dimensional (2D) images [2–10]. Mortensen and Barrett[2, 3] proposed a well-known method called "Intelligent Scissors" which is referred as IS or the original IS from now on. It is to select a route between two points manually selected along which a cost is minimized using dynamic programming (DP). There are several improved methods based on IS[4, 5]. But all these require manually selected initial and terminal points.

There is a well-known method called "Snakes" by Kass et al.[6] which defines a boundary as an energy minimizing problem interactively starting from an initial boundary that is set manually set or some cases based on a priori known knowledge. The energy is defined as a combination of three terms: Ngoi and Jia[8] proposed a method for extracting an object region using Snakes with color information. But again this method requires an initial boundary manually selected which is reasonably close to the final one.

J. Bigun and T. Gustavsson (Eds.): SCIA 2003, LNCS 2749, pp. 1130–1137, 2003.

Mitsunaga et al.[10] proposed an object boundary extraction method based on the gradient of the color image and the alpha value which they define. It generates a contour line interpolated with cubic splines. There are many other works based on edges and boundaries from statistical viewpoints.

In this paper we propose a new method like IS which, however, does not require two manually set points: starting and terminal points. Snakes based methods require an initial approximate boundary and tend to provide final boundaries which are smooth. But the external perimeters of flowers can be sometimes very sharp or zigzag. So we decided to explore IS-like methods that would be able to deal with this situation.

We assume here that an object is placed at or near the center of image. It is focused sufficiently well so that the object of interest has sharper external edges than other objects in the background. However its proportion with respect to the total image size is not known. Also its color is not known. We should find a method of extracting an object region from such a 2D image automatically.

2 Proposed Method

2.1 Outline of IS

See [2] for details of IS. Here we describe its essence. This is a method to determine a route between two points manually entered that minimizes a cost. The defined cost is the sum of local costs defined by six weighted terms:

1. Laplacian zero-crossing (f_Z)
2. Gradient magnitude (f_G)
3. Gradient direction (f_D)
4. Edge pixel value (f_P)
5. Inside pixel value (f_I)
6. Outside pixel value (f_O)

Hence, the local cost is given as:

$$l(\boldsymbol{p},\boldsymbol{q}) = w_Z \cdot f_Z(\boldsymbol{q}) + w_G \cdot f_G(\boldsymbol{q}) + w_D \cdot f_D(\boldsymbol{p},\boldsymbol{q}) + w_P \cdot f_P(\boldsymbol{q}) + w_I \cdot f_I(\boldsymbol{q}) + w_O \cdot f_O(\boldsymbol{q})$$

Where w_Z, w_G, w_D, w_P, w_I, and w_O are weights. We set $w_Z = 0.3$, $w_G = 0.3$, $w_D = 0.1$, $w_P = 0.1$, $w_I = 0.1$, and $w_O = 0.1$ which are identical with those of IS.

2.2 Performance of IS

After computing local costs on the entire image in advance, then the process of computing a route is very fast with IS.

Consider an image shown in Figure 1. It is a photograph of a dandelion. Figure 1(a) shows the computed route starting from point A and ending at point B with applying the (original) IS method. This route is not satisfactory. We investigated why this route is selected instead of the exact or approximate perimeter. The main reason is that IS tends to prefer a route with a shorter path rather than a longer path due to the fact that the total accumulated cost can be minimized with a shorter path.

The next trial was to manually enter 31 points whose consecutive neighboring pair can be the initial and terminal points. Then IS could provide a satisfactory result as shown in Figure 1(b). But this does not meet our condition of 'automatic method' since it requires human interactions.

Another important finding is that the total accumulated cost (=14425) of the original IS is significantly lower than that (=38537) with 31 manually selected points. The main reason is that the route length of the latter is about 3.2 times longer than the former (see Table 1). To solve this problem, we then computed the average cost per length or the unit distance where the city block distance is used. It is 64.1 vs. 54.0. Therefore the route selection should not be based on the accumulated cost but rather on the average cost.

(a) Using two points. (b) Using 31 points.

Fig. 1. Connected between ***A*** and ***B*** by IS.

Table 1. Cost of IS.

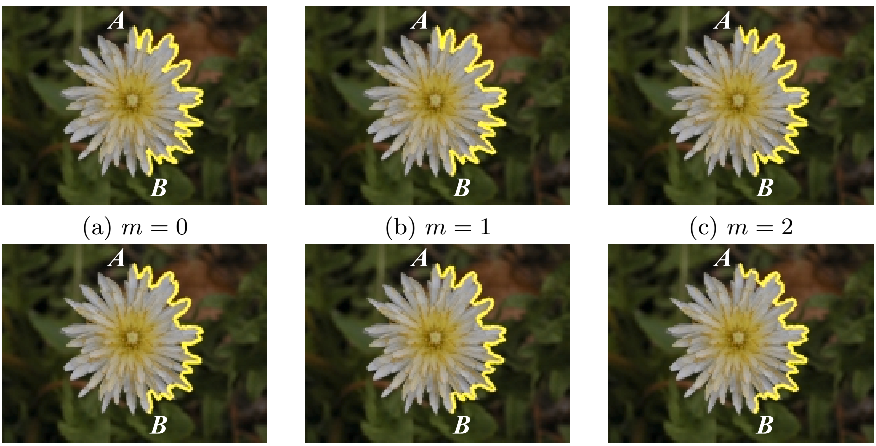
given points	route length	accumulated cost	average cost	figure
2	225	14425	64.1	Fig.1(a)
31	713	38537	54.0	Fig.1(b)

2.3 New proposed method

Our idea is to consider only a partial average cost from all the past pixels within a distance of m from the coordinate of the current pixel. The method with $m = 0$ means that no past pixel is used for determining the future route; the next pixel to be selected is the one having the smallest cost among seven directions. (The past direction is not selected by checking the list of past pixels.) The method with $m = 1$ considers (the sum of the past one pixel cost and the next cost)/2 to select which direction to move. The route with the lowest associated cost is selected among many alternative routes from two end points like IS.

Figures 2(a) to (f) show the results for m = 0, 1, 2, 5, 10, and 50. And the resulting accumulated and average costs are listed in Table 2. The average costs with different m are very similar. However, we decided to choose $m = 1$ for our application because the resulting computed route with $m = 1$ agrees best with the manually determined route. It is expected that a larger m might be preferred for slowly changing boundaries.

Although IS is a dynamic programming (DP) method, our method is not DP since the cost is not additive (or cumulative). We will call our proposed method the Route Tracing (RT) Method. Let RT_m represent our RT method using all the pixels within a distance of m from the current pixel.



(a) $m = 0$ (b) $m = 1$ (c) $m = 2$

(d) $m = 5$ (e) $m = 10$ (f) $m = 50$

Fig. 2. Connected between ***A*** and ***B*** by RT_m.

Table 2. Cost of RT_m.

m	route length	accumulated cost	average cost	figure
0	1204	41942	34.8	Fig.2(a)
1	1328	45403	34.2	Fig.2(b)
2	1231	42107	34.2	Fig.2(c)
3	1191	40477	34.0	
5	1132	38325	33.8	Fig.2(d)
10	1110	37827	34.1	Fig.2(e)
50	1027	35181	34.3	Fig.2(f)
100	1035	34682	33.5	

3 Automatic Extraction

3.1 The First Search

Our next problem is to extract an object region from a photograph without manually entered initial and terminal points. We first tried RT_1 with the upper and lower mid points (200,0) and (200,299) as the initial and terminal points where the origin (0,0) is set at the left, top corner for an image of 400×300 pixels. However, there were cases that objects were completely circumvented as shown Figure 3(a). Therefore our current choice is three points P_0, P_1, and P_2 by adding the center point (200, 150). Actually we execute RT_1 from P_1 to P_0 and then from P_2 to P_0 separately as shown Figure 3(a) and (b). Then we plot the local cost, $c(l), l = 1, 2, \cdots, N$, where N is the number of pixels along the selected route.

3.2 The Second Search

Then we identify one most probable edge point from each cost profile as shown Figure 4. For this task, we observe a histogram of local costs. Then we select a 10% level t_1 as the threshold. (This means that at least 10% of external edges are traced.) The first points (C_1 and C_2) from P_1 to P_0 and P_2 to P_0 lower than t_1 are selected as the initial and the terminal point for RT_1.

3.3 The Third Search for a Closed Loop

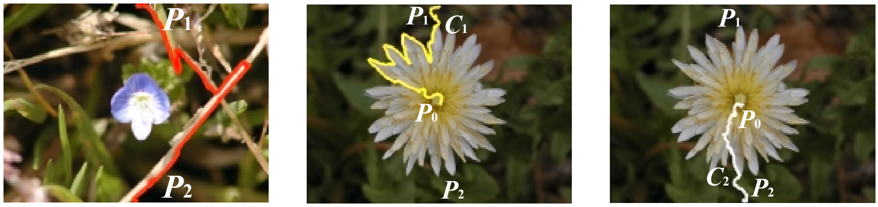
Actually RT_1 is executed two times to obtain a closed loop. After the first execution is done to obtain Route 1, then this route is prohibited for the second execution. Actually, the local cost of pixels within a distance of 3 from the Route 1 is set to be very high. Then we execute RT_1 again to obtain Route 2 which is usually on the other side. The resulting routes of Route 1 and 2 will always form a loop as shown Figure 5.

3.4 The Final Check

Figure 5 shows a typical example of a closed loop obtained in the previous section. A segment at the upper part of the closed loop runs in the background. In order to make sure, we consider the color difference profile $d(l)$ between two pixels inside and outside along a route. If it is in the background or inside the flower region, the color difference becomes lower. Figure 6(b) shows the color difference profile along the same route of Figure 6(a). In order to account for the local cost and color difference, we define a new composite function $f(l)$ (see Figure 6(c)) as:

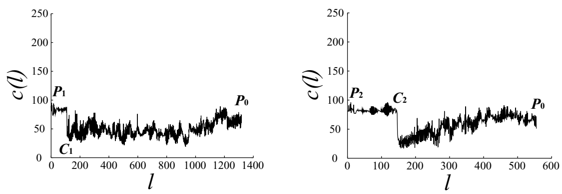
$$f(l) = w_c \cdot \big(255 - c(l)\big) + w_d \cdot d(l)$$

Experimentally we have chosen weights $w_c = 0.45, w_d = 0.55$ and a threshold $t_2 = 280$. With this threshold, two points on the perimeter are identified. Then RT algorithm is executed with two points as the initial and terminal points. Figure 7 shows the final result.



(a)$P_1 \rightarrow P_2$ (circumvented) (b)$P_1 \rightarrow P_0$ (c) $P_2 \rightarrow P_0$

Fig. 3. First search.



(a) $P_1 \rightarrow P_0$ (b) $P_2 \rightarrow P_0$

Fig. 4. Cost profiles.

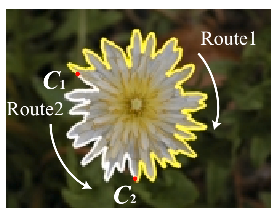
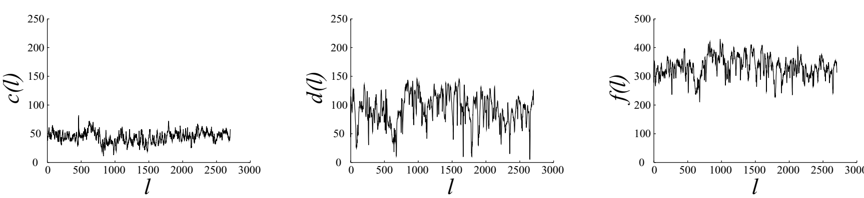


Fig. 5. Third search (closed loop).



(a) $c(l)$ (b) $d(l)$ (c) $f(l)$

Fig. 6. Profiles of closed loop.

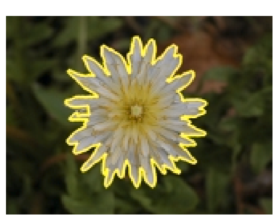


Fig. 7. Extracted result.

4 Experiments

We collected 200 photographs of 400 × 300 pixels. We classified them into four types. Type I and II contain pictures of well-focused objects with well defocused background. But object's perimeter of Type I is slowly changing, while that of Type II is frequently changing primarily because the object contains many petals. Type III is a group of pictures that contain two or more separated flowers. Usually only one of them is focused. In Type IV, some of flowers are overlapping or behind the object flower. Among 200 pictures, 58, 81, 47, and 14 pictures are classified into Type I, II, III, and IV, respectively.

To evaluate the performance of our method, we introduce the following criteria:

α: successful if the correct percentage of the route per image is more than 95%. Otherwise unsuccessful.

β: successful if the correct percentage of the route per image is more than 90%. Otherwise unsuccessful.

γ: successful as to the correct percentage per image.

Criterion γ means that if 70% of the perimeter is correctly routed, we give a 70% correct rate for the image. Table 3 lists the successful percentage for the four types with criteria α, β and γ. Figure 8 shows extracted results of some photographs.

Table 3. Extracted results of all images.

type	the number of images	α[%]	β[%]	γ[%]
Type I	58	96.6	96.6	93.6
Type II	81	91.4	92.6	90.8
Type III	47	89.4	91.5	92.7
Type IV	14	85.7	92.9	90.5
total	200	90.7	93.4	91.9

5 Conclusion

We presented a new automatic method for extracting an object region from a photograph where the object is placed in the middle. We analyzed the performance of a well-known method "Intelligent Scissors" and found that the additive cost used is the main source of the difficulty. We proposed a new automatic method called the RT (Route Tracing) method which employs an averaging cost. Its performance was evaluated experimentally with 200 images. It was found that the RT provided a correct rate of about 93% for images with well-focused objects and well-defocused background.

References

1. Takeshi Saitoh and Toyohisa Kaneko. Automatic recognition of wild flowers. *IEICE (Japan)*, J84-D-II(7):1419–1429, Jul. 2001.

Fig. 8. Extracted results.

2. Eric N. Mortensen and William A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384, September 1998.
3. Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. *Computer Graphics, SIGGRAPH 95 Conference Proceedings*, pages 191–198, August 1995.
4. Detlev Stalling and Hans-Chrisitian Hege. Intelligent scissors for medical image segmentation. *Proc. of 4th Freiburger Workshop Digitale Bildverarbeitung in der Medizin*, pages 32–36, 1996.
5. Mark A. Ruzon and Carlo Tomasi. Alpha estimation in natural images. *IEEE Conf. on Computer Vision and Pattern Recognition 2000*, pages 18–25, 2000.
6. Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1998.
7. Amir A. Amini, Terry E. Weymouth, and Ramesh C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(2):855–866, September 1990.
8. K.P.Ngoi and J.C.Jia. An active contour model for colour region extraction in natural scenes. *Image and Vision Computing*, (17):955–966, 1999.
9. Jaesang Park and James M. Keller. Snakes on the watershed. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(10):1201–1205, October 2001.
10. Tomoo Mitsunaga, Taku Yokoyama, and Takashi Totsuka. Autokey: Human assisted key extraction. *Computer Graphics, SIGGRAPH 95 Conference Proceedings*, pages 265–272, August 1995.