Performance Improvement of Leaf Contour Extraction in Complex Background for Leaf Recognition

Kue-Bum Lee, Xianghua Li and Kwang-Seok Hong

School of Information and Communication Engineering, Sungkyunkwan University, 300, Chunchun-dong, Jangan-gu, Suwon, KyungKi-do, 440-746, Korea
Leo0608@skku.edu, gidghk88@skku.edu, kshong@skku.edu

Abstract

In this system we improved the performance of an automatic system for extracting leaf contour. The proposed leaf contour extraction method consists of three major procedures: the detection of four edge points, and contour tracing. Leaf detection includes two stages: feature extraction and matching. For the leaf contour extraction part, we present a new technique for automatically identifying the contour of a leaf in an image. For contour tracing, an Intelligent Scissor (IS) algorithm is applied. The color gradient magnitude and Canny edge detection are analyzed and included as the cost terms of the Intelligent Scissor algorithm. The color gradient magnitude cost term is implemented so that it can act directly on the three components of the color image. For the third procedure, we implement of the performance improvement. The main idea about proposed method is when the program detected the incorrect four edge points, we using the mouse and clicked the right position instead of the incorrect one.

Keywords: *Intelligent Scissor (IS) algorithm, Leaf Contour Extraction, Edge Point, color gradient magnitude*

1. Introduction

When we wander around fields, we can find a number of wild plants around campus and park. But we rarely know their names. We may consult with a book on wild plants or search the internet, but even if such a book or smart phone is on hand, it is not an easy task to find a proper section or an exact page showing the plant. It would be useful if we direct a digital camera to a plant and then an algorithm implemented in a notebook PC will recognize the plant and tell its name. The first task of recognition is to extract the object of interest from the complex background.

There are some existing methods proposed for the problem of extracting an object region from two dimensional images. Previous work [1] used an approach to avoid the difficulty by physically picking up a flower and placing it on a black sheet before taking a photograph. This method is undesirable from an environmental viewpoint in addition to some laborious work involved prior to photographing. Some interactive methods such as computed-assisted interactive recognition (CAVIAR) [2] have been developed [3] to exploit human perception. A rose-curve was generated for the test flower, and the top three candidates were proposed to the user who can then either select the right flower (if present) or modify the rose-curve to obtain a new set of propositions. This is done iteratively until the right flower is found, which is not suitable for our application because of the multiple user interactions.

And the other automatic method was developed under the assumption that the flower is focused while the background is defocused. The method then uses a normalized cost (NC) [4] method, which needs a manual entry point on the boundary. They overcame this drawback by implementing an automatic method that minimizes the NC among a set of carefully chosen entry points.

The main novelty of this paper lies in that the proposed robust method is without any a priori assumption and initial contour information. First, we use a robust method to detect four edge points on the leaf contour [5]. Then, we employ the well-known method Intelligent Scissor (IS) algorithm [6] to connect the whole leaf contour. For enhancing the edge accuracy, we also use color gradient magnitude to compute the image gradient directly. The result of the experiments shows that the algorithm presented in this paper works well on many different types of leaves.

2. Leaf Area Extraction

In order to extract the leaf contour, the proposed leaf contour extraction algorithm involves three major stages that detection of leaf edge points, contour tracing using the IS algorithm and the mouse click section if the program detected the incorrect edge points. We discuss the process of detecting leaf contour in detail below.

2.1. Leaf Contour Extraction

For extracting an object contour, there is a well-known method by Mortensen and Barrett called IS [5]. It is a manual method to draw a contour based upon a number of manually selected points on the visually identified contour, providing a route that minimizes the sum of local costs. It is fast because the principle of dynamic programming is employed. In order to extract the leaf contour as correctly as possible, the proposed system we regard one point as a boarder of an object if the gradient of that point is greater than the threshold value in advance. We use two lines, one horizontal and one vertical line to cross the leaf. By detecting the location with larger change in gradient magnitude on two lines, we can determine four leaf edge points. First we define local cost which associated with every pixel on each scan line. The local cost (LC) of a pixel on a scan line is defined as follows:

$$LC = 1 + MG - G \tag{1}$$

Where, G denotes the gradient magnitude for the pixel and MG denotes the maximum gradient magnitude of all pixels in the image. According to the definition of local cost, we can see that a pixel with strong edge gradient will have a small local cost. In this study, the Sobel were employed to compute the horizontal and vertical gradient magnitudes of a pixel. Let $I_R(x,y)$, $I_G(x,y)$, and $I_B(x,y)$ denote respectively the R, G, and B color values of a pixel locating at (x,y). For R color value, the corresponding gradient magnitude, notated $G_R(x,y)$ is defined as follows:

$$G_R(x,y) = \sqrt{G_{R,H}^2(x,y) + G_{R,V}^2(x,y)} \tag{2}$$

Where, $G_{R,H}(x,y)$ and $G_{R,V}(x,y)$ denote respectively the horizontal and vertical gradients derived by convolving the horizontal and vertical Sobel operators with the 3×3 image blocks centered at (x,y) and can be described by the following equations:

$$G_{R,H}(x,y) = \left|I_R(x+1,y-1) + 2I_R(x+1,y) + I_R(x+1,y+1)\right| - \left|I_R(x-1,y-1) + 2I_R(x-1,y) + I_R(x-1,y+1)\right| \quad (3)$$

$$G_{R,V}(x,y) = \left|I_R(x-1,y+1) + 2I_R(x,y+1) + I_R(x+1,y+1)\right| - \left|I_R(x-1,y-1) + 2I_R(x,y-1) + I_R(x+1,y-1)\right| \quad (4)$$

The gradient magnitudes for G and B color values, notated $G_G(x,y)$ and $G_B(x,y)$, can be computed in a similar manner. The overall gradient is then defined as the maximum value among $G_R(x,y)$, $G_G(x,y)$ and $G_B(x,y)$:

$$G(x,y) = MAX\{G_R(x,y), G_G(x,y), G_B(x,y)\} \quad (5)$$

According to the previous method mentioned above we using LC is very effective when the image background is relatively simple. However, in a more general case, flowers are often surrounded by leaves and stems and other complex background filler that only a larger change of LC cannot provide a satisfactory result. According the computed local cost associated with each pixel, four profiles of local costs along the lines starting from every Pi (i=1, 2, 3, 4) to P0 are generated. In this study, the estimated stamen region will be excluded on each profile (see in Figure 1.).

Figure 1. Four Edge Points Detection.

These four leaf edge points will be taken as the starting/ending points of the leaf contour tracing algorithm. After determining four edge points, the rest of the leaf contour is automatically traced. In our application, we constructed the cost function of IS that uses Canny edge cost terms and color gradient cost terms. Each pair of the four sets of edge points, (p1 p2), (p2 p3), (p3 p4), and (p4 p1), served as the starting and ending tracing points. Then, we selected the shortest path with minimum energy costs to track the route. The important part of the IS algorithm is how to construct its cost function. In an ideal case, it should contain all image components that have an impact on the position of the leaf contour. However, in our application, because the edge of the petals has a sharp direction, we only use canny edge cost terms and gradient magnitude terms to construct the cost function instead of the traditional IS algorithm which also includes gradient direction cost terms.

The cost function $C(p,q)$ from pixel p to a neighboring pixel q is defined as

$$C(p,q) = \omega_C f_C(q) + \omega_G f_G(q) \quad (6)$$

Where, $f_C(q)$ and $f_G(q)$ represent the Canny edge detection and gradient magnitude cost terms, respectively. Each cost term is weighted by a corresponding weight constant ω_C and ω_G to allow cost terms to contribute to the total cost at different rates.

The local cost term for the gradient magnitude of pixel q is defined as

$$f_G(q) = 1 - G(q) / max(g) \quad (7)$$

Where, $G(q)$ denotes the gradient magnitude in q and $max(g)$ denotes the maximum gradient magnitude of all pixels from the entire image.

2.2. The Incorrect Detected Edge Points Clicked by Mouse

There are lots of incorrect edge points detected by this algorithm. In this part we present the mouse click method to resolve this problem to get the high detection performance.

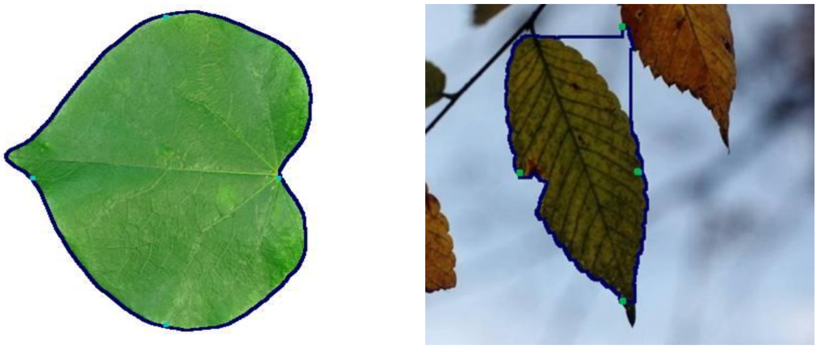
Like above problem, we proposed the mouse clicked method. The main method is draws the horizontal and vertical line on the leaf image and if the edge point is detected on the incorrect location. Then we use mouse and clicked the right position that instead of the incorrect one. In this part, we must choose the right edge point that which is the point using mouse. First, we draw the two yellow lines if the leaf's edge points are detected incorrect position.

Then we used mouse and clicked the right position close to the line. We implemented the function when the point clicked away from line, it will calculate the clicked point coordinate. Then, we calculate the length from the clicked point to the vertical and horizontal line. Which length is shortest and the clicked point will fixed to vertical or horizontal line.

There are also existed no edge points detected any more caused by the gradient value have little changed. No edge points detected result to the fault contour tracing. We clicked four edge points in the yellow line and detect contour again. The result image shows that it can detect the correct contour.

3. Experiments and Results

The proposed automatic leaf contour extraction and mouse clicked system is implemented using Microsoft Visual Studio 2008-supported C++ using the Microsoft.Net Compact Framework 2.0. Figure 2 shows (a) the correct leaf contour detection result, and (b) the incorrect edge point detected result. In figure 2 (b), it has one point is detected in incorrect position. We only clicked one point and instead of the incorrect point and tracing the contour again. Figure 3 shows the contour tracing result using mouse clicked.



(a) (b)

Figure 2. Leaf Contour Detection Result

Figure 3. The Correct Leaf Contour Detection Result

4. Conclusions

In this paper, we present an automatic leaf contour detection and mouse click system. First, in leaf contour auto-extraction part, we are not only detecting the four edge points by local cost terms but also improving the algorithm by adding diffusion, a length constraint and a color constraint. In the tracing phase, we use a derivative of a vector field method to compute the color image gradient directly. Then, the modified IS method is employed to track the contour. The main idea about proposed method is when the program detected the incorrect four edge points, we using the mouse and clicked the right position instead of the incorrect one. In the experimental results, the performance has been proved to the point that our algorithm has robust automatic contour extraction for many different types of images.

Future work is necessary to research more ideas are thrown that could improve or complete the current system. In addition, we will experiment using a many database of leaf image.

Acknowledgements

This research was supported by MKE, Korea under ITRC NIPA-2012-(H0301-12-3001) and PRCP through NRF of Korea, funded by MEST(2011-0018397).

References

[1] T. Saitoh and T. Kaneko, "Automatic recognition of wild flowers", Proceedings of the 15th International Conference on Pattern Recognition, **(2000)** September 3-7; Barcelona, Spain.

[2] G. Nagy and J. Zou, "Interactive visual pattern recognition", Proceedings of the 15th International Conference on Pattern Recognition, **(2002)** August 11-15; QC, Canada.

[3] J. Zou and G. Nagy, "Evaluation of model-based interactive ower recognition", Proceedings of the 15th International Conference on Pattern Recognition, **(2004)** August 23-26; Cambridge, UK.

[4] T. Saitoh, K. Aoki, T. Kaneko, "Automatic extraction of object region from photographs", Proceedings of the 13th Scandinavian Conference on Image Analysis, **(2003)** July.

[5] I. K Park, B. H Ahn AND G. S Choi, "An Edge Detection for Face Feature Extraction using λ- Fuzzy Measure", The Journal of IWIT (The Institute of Webcasting, Internet and Telecommunication), 9, 4 **(2009)**.

[6] E. N. Mortensen and W. A. Barrett, "Intelligentscissors for image composition", Proceedings of the SIGGRAPH, **(1995)** August.

Authors

Kue-Bum Lee received his B. S. in computer science from the Seoul Health College, Sungnam, Korea, in 2006, and M.S. in electronic and electrical engineering from Sungkyunkwan University, Seoul, Korea, in 2008, respectively. He is presently a Ph.D candidate at the department of Information and Communication Engineering Sungkyunkwan University.

Xianghua Li received the B.S. in computer engineering from the Dalian Nationalities University, Dalian, China, in 2010 respectively. From September 2010, she study for master course in the Sungkyunkwan University, Seoul, Korea, until August 2012.

Kwang-Seok Hong received his B.S., M.S., and Ph.D. in Electronic Engineering from Sungkyunkwan University, Seoul, Korea in 1985, 1988, and 1992, respectively. Since March 1995, he has been a professor at Sungkyunkwan University, Suwon, Korea. His current research focuses on human-computer interaction, five-sense recognition, interaction, and representation.