2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Face Contour Detection by Using B-spline Snake for Creating Human Face Caricature

Jang-Hee Lee, Jae-Kun Woo, Hoon Kang
School of Electrical and Electronic Engineering, Chung-Ang University
flyigace@sirius.cie.cau.ac.kr

Abstract - This paper deals with the making avatar like a caricature from human face image which is made by web camera. Generally, the Image made by web camera is not low quality but also, there are always various lights and backgrounds. So, It is impossible to recognize a human face's contour by some methods which only find some feature points of a image. Therefore, In this paper, we propose a new method for overcoming defeat of that methods. First, we got the area of human face roughly by color information. And then, we could find the exact human face's contour by using B-spline Snake.

1. Introduction

The existing creation method of a avatar is to combine facial component templates like eyes, lip, nose, and etc. But, This method can't reflect user's external features well. So, It can't include the identity of user. The purpose of this paper is to overcome this limit of existing creation methods. In this paper, as it creates the avatar using the user's photo, it reflects user's identity sufficiently. And this created avatar has feature looks like user's caricature.

For this work, the background and the face need to be segmented, but there are two problems. First, the face detection is very sensitive about the light variation. The other, it is very hard to search the exact face contour. As the solution of former, we used the HSI color model. Generally, It is used the RGB color model in computer. But, The RGB color model varies with the light's intensity sensitively. On the other hand, the HSI color model is robust against the variation of light. So, we used HSI color model. As the solution of later, we used the B-spline Snake. It guarantees smoothness and continuity of object contour, because B-spline Snake is a method of using Prior Knowledge progressively like a recognition process of human. With this method, we can protect that the detected image is damaged by a light or a shadow. And what's more, It is easy to implement by the computer, Because this method is organized by the linear algebra. Recently, there are many studies about the B-spline Snake in a image recognition field because of these reasons. We segment the facial area and the background using the detected facial contour. And for a effect like a drawing, we used Modified Laplacian of Gaussian Filter to obtain edge image in face

2. Face Region Detection

2.1 RGB model vs. HSI model

Generally the stored image in computer is represented with the RGB color model. Most defect of this RGB color model in image recognition are that each component contain the light's intensity, so each componential value is varied largely by the light variation. There are frequent occasions not only that light is distributed irregularly, but also that target is shaded by the light. The image consisted of the RGB color model converted the HSI color model for robustness against the light variation better because of the above situation. If we use the HSI color model, we can have a independence from shadow's or light's effect somewhat. Most of all, the Intensity can be regarded as one independent component. So, if we neglect intensity, we can design robust algorithm against the light variation. Fig 1. represents about this HSI model.

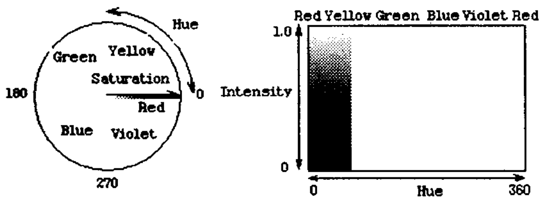


Fig 1. HSI Color Model

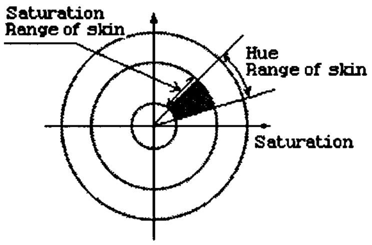


Fig 2. Skin color range at Hue-Saturation Space

2.2 Face Color Region Definition in Hue-Saturation Space

The Detecting facial area is started with defining a region which is suitable as a skin area in the Hue-Saturation space. Actually, it is impossible that skin area can be defined as one region in the Hue-Saturation space due to various lights and environments[2]. So, we determined to obtain a user's input. First, we receive the tip of nose from user by the mouse click. If we receive the location of nose, we can define the skin area by this information. Assuming received point is (x, y), we investigate total 30*30 pixel's Hue and Saturation from (x-15, y-15) to (x+15, y+15). Then, we can determine region of skin color. Fig 2. represents the determined area at Hue-Saturation space. Then, We detect facial area by applying Flood Fill Algorithm at this region which searched as the skin color range

2.3 Face Region Detection

Using the skin color range which is found by preceding part, we could detect skin region. We used the Modified Flood Fill Algorithm which named 'paint box'. In this paper, modified flood fill algorithm investigates whether neighbored value is skin range or not. Fig 3 is a result image by using this algorithm. But it is impossible to segment a chin line. So, we used the B-spline snake.

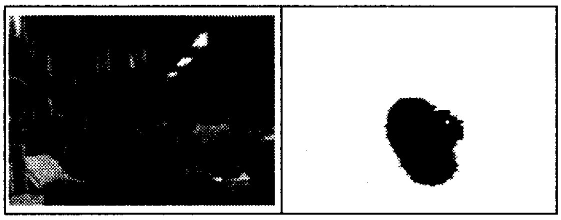


Fig 3. Result by Using Modified Flood Fill Algorithm

3. B-Spline Snake

B-spline Snake is the method that it can represent the curve effectively in computer graphic. This curve constructed as weighted sum of Basis function[1].

$$x(s) = \sum_{n=0}^{N_B-1} x_n B_n(s) \qquad (1)$$

$$r(s) = U(s)\,Q \qquad \text{for } 0 \le s < L \qquad (2)$$

where Q :a control vector

$U(s)$: a vector of B-spline basis function

L : a interval length

N_B: the number of control point

3.1 The Curve Fitting Algorithm

Fig 4. shows the result of drawing using Q_x, Q_y and the basis function on the input image. First of all, after receiving user's input about location of nose, we measured an area of face using the result of Fig 3. Next we scaled a facial template and moved the facial

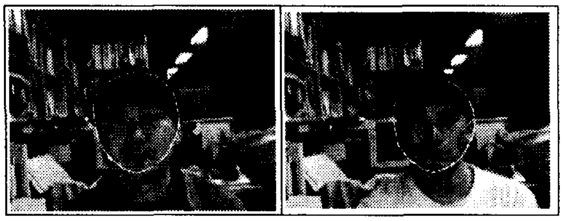


Fig 4. Setting Up of Initial Shape

template with setting on the basis of nose. we could obtain Fig 4. Fig 5. is the image which is sampled by normal vector direction at each control point.

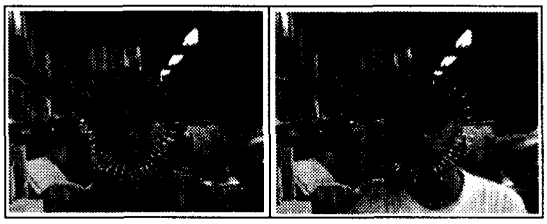


Fig 5. The Orientation of Normal Vector at Control Point

In this procedure, the orientation of the normal vector is obtained by using the inner product. The next is procedure to obtain Q_f. In this procedure, a filter shown in Fig 6. is used about each sample line.

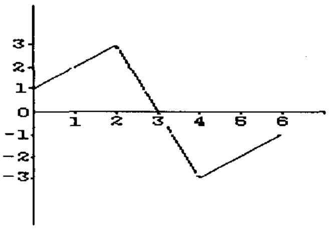


Fig 6. Edge Detection Filter

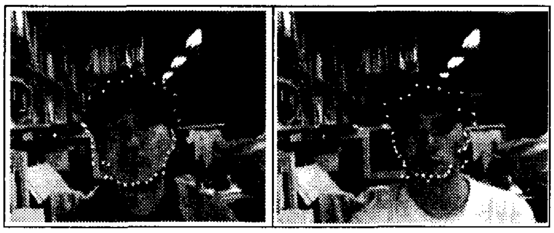


Fig 7. Result Image Obtained Q_f

If the sample convoluted with this filter, the place where has extreme difference between the pixels' brightness represented the highest value. Fig 7 is the Result which is obtained by representing the maximum of convoluted values.

As it is confirmed by Fig 7, many points are located at the desired place. But a few point are located at the undesired place. This appearance can be explained that complex background has the maximum difference of brightness value than area which segment the face and the background. Consequently, we must correct this error on the basis of the detected point. Fig 8. show the result to be applied Curve Fitting Algorithm as below[1].

※ Recursive Algorithm for Curve Fitting

```
1. choose th sample.
   $s_{i,} i=1,2,\ldots,N$ s.t. $s_1=0, s_{i+1}=s_i+h, s_N=L$
2. for each i, with orientation of $\overline{r}(s_i)$
   search the location of $r_f(s_i)$ by using
   proper image processing filer.
3. initialize with $\mathbf{Z}_0=0, S_0=0$.
4. for $i=1,2,\ldots,N$ , iteration below
   formula
       $\nu_i=(r_f(s_i)-\overline{r}(s_i))\cdot\overline{n}(s_i)$
       $\mathbf{h}(s_i)^{T=}\overline{n}(s_i)^T U(s_i) W$
       $S_i=S_{i-1}+\frac{1}{\sigma^2}\mathbf{h}(s_i)\mathbf{h}(s_i)^T$
       $\mathbf{Z}_{\mathrm{i}}=\mathbf{Z}_{\mathrm{i-i}}+\frac{1}{\sigma^2}\mathbf{h}(s_i)\nu_i$
5. $\mathbf{Z}=\mathbf{Z}_{\mathrm{N}}, S=S_N$
6. calculate $\widehat{X}$.
       $\widehat{X}=\overline{X}+(\overline{S}+S)^{-1}\mathbf{Z}$
```

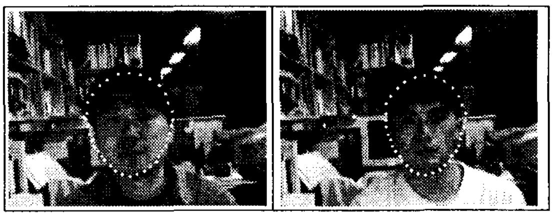
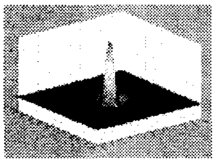


Fig 8. Result Obtained by Correcting Curve

4 Edge Detection

Observing a painters' caricature, they have a tendency to accentuate the facial contour or boundary of component by painting a dark color at that region. So, we need image that has the only edge for drawing caricature. There are representative filter for detecting the edge like Sobel filter, Robert filter, Canny filter, etc. These filters use size of 3x3 due to efficiency of speed. But edge image obtained from these method has rough result. On the other hand, the Laplacian of Gaussian filter can be increased the size arbitrarily.

At actual computer operation, it is used the mask that can be implemented simply. It is obtained easy by operating a convolution mask and image. Fig 9. shows the mask of Laplacian of Gaussian the size of 5x5



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Fig 9. Laplacian of Gaussian Mask

5. Final Result

It is created that the detected facial region is added the edge image as the final result. This is the final result of Avatar.

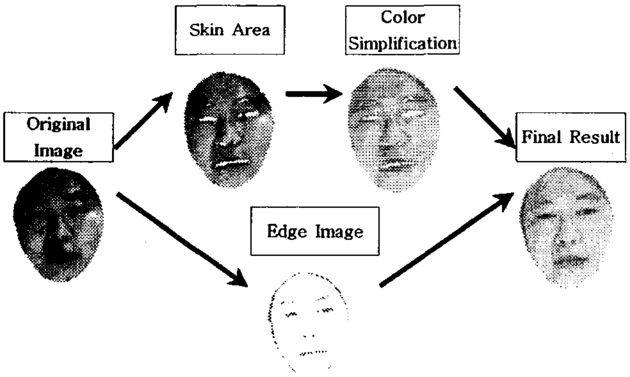


Fig 10. Final Result

6. Conclusion

The existing creation methods of avatar which combine the facial component templates can't reflect user's external features well. To solve this problem, a new method is proposed in this paper. One is to detect the area of human face roughly by color information. And The other is to find the exact human face's contour by using B-spline Snake. In respect of detecting an approximate face region by using color information, there is a problem that the detected face is affected by the light variation. But this problem could be solved by using not the RGB color model but the HSI color model. Also, There is a problem that the exact face contour can be extracted by the only color information. but, nearly exact face contour could be extracted by using the B-spline Snake. We didn't deal with statistical analysis about Relation between detection ability of face region and the complexity of the background, the light variation. so, it need to research more.

Reference

[1] Andrew Blake, Michalel Isard, *Active Contours*, Springer, 2000

[2] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, second edition, Prentice Hall, 2002

[3] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, *Machine Vision*, McGraw-Hill, 1995

[4] Harry Wechsler, P.Jonathon Phillips, *Face Recognition From theory to Applications*, Springer, 1998

[5] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, *Computer Graphics Principles and Practice*, Addison-Wesley, 1997

[6] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical qRecipes in C++*, CAMBRIDGE UNIVERSITY PRESS, 2002