Fast Multi-Face Detection Using Facial Component Based Validation by Fuzzy Logic

Pham The Bao
ptbao@{mathdep.hcmuns.edu.vn, care2.com}
Mathematics & Computer Sciences Department
University of Natural Sciences
Hochiminh City, VietNam

Abstract - *A human skin color range is used to find the candidates by segmentation and the rules of fuzzy factor are established by using facial characteristics to determine whether the candidate is face or not. We have separated two connected faces. In this paper, after segmentation with skin color, facial components are used instead of facial features to build fuzzy factors for verifying face candidates. Concurrently with detection, we separate connected candidates by Fast Marching but a fuzzy factor is built to choose radius which depends on the size of a candidate and the limit of spread. We use probability of rules and calculate probability of face. Experimental results demonstrate a successful face detection over a wide range of facial variations in color, position, scale, orientation, 3D pose, and expression in images from several photo collections (both indoors and outdoors). The computation time in this study is the same as that of the previous research, but the failure rate in the proposed algorithm is lower than that of the previous method.*

Keywords: Face detection, Fuzzy logic, Fast Marching.

1 Introduction

Images containing faces are essential to intelligent vision-based human computer interaction, and research efforts in face processing include face recognition, face tracking, pose estimation, and expression recognition, etc. However, many reported methods assume that the faces in an image or an image sequence have been identified and localized. To build fully automated systems that analyze the information contained in face images, robust and efficient face detection algorithms are required. Given an image, the goal of face detection is to identify all image regions which contain a face regardless of its three-dimensional position, orientation, and lighting conditions. Such a problem is challenging because faces are not rigid and have a high degree of variability in size, shape, color, and texture.

Numerous techniques have been developed to detect faces in a single image [2]. A few methods detect multi-face [3] and most methods don't solve the problems with connected faces. Most face detection methods are knowledge-based, feature-based[4], template matching, neural network, or their combination. Among them, feature-based methods provide a good solution for detecting faces in different poses and sizes. Particularly, color-based approaches are useful cues for face detection [5]. However, these approaches face difficulties in robustly detecting skin colors in the presence of complex background and different lighting conditions.

We propose a novel algorithm for automatic and fast detection of human faces that is able to handle a wide range of variations in static color images. Based on the compact distribution of human skin color in YCbCr color space and characteristics of eyes and mouth color [5, 6] to segment, our new method detects skin regions over the entire image, and then generates face candidates based on these skin regions' borders. The algorithm uses a fuzzy controller with fuzzy factors such as the relationship of face, eyes and mouth on a face [7, 8, 9, 10], with the relationship we don't heed the directions of faces. Each rule of fuzzy factors has a probability and weight for verifying each face candidates. In this process, Fast Marching method is used to separate connected face regions. We build the fuzzy factor to choose the coefficients such as radius, the limit of spread supporting Fast Marching method.

2 The Baseline System

We build a system which is based on human skin color segmentation to choose the candidates from images. We don't take all pixels inside of candidates since we choose only the boundaries, as shown in Fig. 1. The reason of treating only the pixels on boundaries is to make the computation time as short as possible.

2.1 General Algorithm

We suggest the algorithm which has 06 steps to detect. Given an input color image, our face detection algorithm is as follows:

Step 1: Find skin pixels and nonskin pixels, then we get a skin image (binary image).

Step 2: Remove noises from the skin image by opening and closing operator.

Step 3: Find border pixels for generating face candidates then we get holes inside.

Step 4: For each face candidate then find 03 holes, consider a rule base for verification:

If its fuzzy values is satisfied then go to Step 5.

Else If its fuzzy values is not satisfied then eliminate it.

Else If its fuzzy values is not sure then go to Step 6.

Step 5: Give the detected face.

Step 6: Consider the following special cases:

If it is a connected region, then separate it by Fast Marching method.

If it does not contain any holes, then find absent features by filtering and thresholding.

Back to step 4.

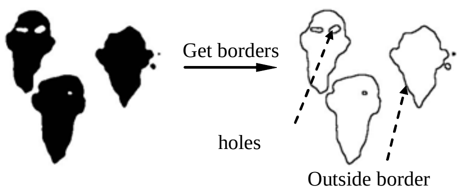

Fig. 1. Borders of the regions.

2.2 Skin color segmentation

In the beginning of our algorithm, the color segmentation is used to classify skin-color pixels and nonskin-color pixels in the input image. There are many ways for this classification [11], but we choose the easiest method. It is based on the fact that skin color is compactly clustered in color spaces such as TSL, YCbCr, and HSV which separate luminance and chrominance components. We adopt the YCbCr space since it is perceptually uniform, and widely used in video compression standards.

A pixel is classified to be a skin pixel if its color values satisfy: $Y_1<Y<Y_2$, $Cb_1<Cb<Cb_2$, $Cr_1<Cr<Cr_2$ [11, 12]. The thresholds are based on skin patches collected from different skin races from the Internet. As a result, a binary image demonstrates this segmentation. Then, an opening and closing operator [13] is used to remove tiny noises in the segmented image.

The next step is finding borders of each skin region. A pixel is on the border of a region if it is a neighbor of both a skin pixel and a nonskin pixel. After having these border pixels, we get holes inside this region. The outside borders which truly enclose the skin regions are face candidates, as shown in Fig. 1. We also notice that holes may be eyes, noses, lips or any noises on the face regions.

Sometimes, we don't find the holes inside the candidates because the candidate is influenced by light conditions, skin tone, etc.…. We use other conditions to determine the holes. We calculate 02 values Crb and Cr2 by the following formula:

$$Crb = Cr/Cb \text{ and } Cr2 = Cr*Cr \tag{1}$$

With eyes, the feature color is brown color, so we use condition Crb>3 and G<100 and B<100 to find the holes which are properly eyes. The mouth feature color is red thus we compare $|Crb|>2$ and Cr2>1300 [5, 6] to find a mouth component of faces. This condition is based on an eye map and a mouth map [2].

3 Facial Component Based Validation

A human face has a wide range of variations in a color image. Therefore, to get a high detection rate and low false positives, we need a flexible face detection algorithm. Fuzzy logic approaches are appropriate. Our face candidate verification process is based on the structure of a fuzzy controller of which fuzzy factors are the positions and rates of components of face. For each candidate, we choose 03 holes which can make a triangle with two possible eyes and a mouth. We build the rules based on the relationship of left eye, right eye, mouth. The DisEye is the distance of left eye and right eye. The HeightEyeMouth is the distance from the point which is the middle of left eye and right eye, to mouth. The DisMouth is the distance from the hole which is a probable mouth to the point which is real mouth, as in Fig. 2. There are the coefficients to build rules.

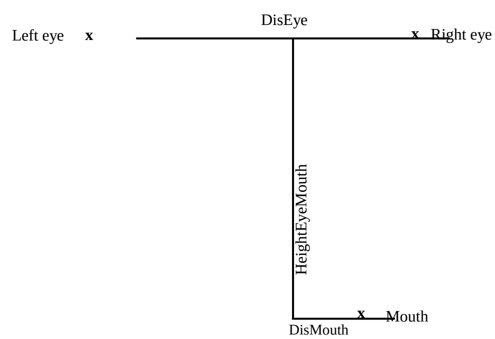

Fig. 2. Position of components

3.1 Separating connected faces

For each candidate, we must know how many holes the candidate has inside and also the size of the candidate. We build a fuzzy factor to determine the radius to spreading and to find how many connected faces this candidate can possibly have. After that, we have enough required coefficients, using fast Marching method to separate connected faces, as shown in Fig. 3. Normally, the maximum number of connected faces making a candidate is about ten.

3.2 Determining components

After segmentation by human skin color, the relationship's components of face can have 03 cases [14].

- Eye regions and mouth region are separated. This is the best case.
- One of two eye regions is connected with the boundary because the effects of pose of face or the light condition.
- Two eye regions are connected. They used eyeglasses or the light condition had an effect.

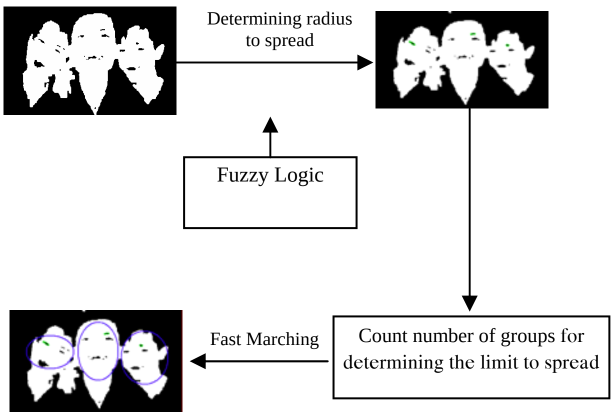

Fig. 3. Separating connected faces

3.2.1 Solving the first case

For the solution in this case, we calculate some values as follows:

- Rate of distance of eyes and width of face.
- Rate of distance from the center of eyes to mouth and width of face.
- Rate of DisMouth value and DisEye/2 value.
- Sizes of LeftRegion, RightRegion, MouthRegion regions.

And we use a fuzzy factor to check whether or not each triple hole is a component of face by applying the rules. Each rule has a value, which is called reliability. We calculate p, which is the average reliability, of the triple hole. With every triple hole we find the maximum of p_i with i=1, 2, …. If p_{max} value is less than 0.65, this triple hole is not the component of face; otherwise, this triple hole is the component of face, as in Fig. 4. Fig. 5 is the example.

3.2.2 Solving the second case

We find the first eye which is not connected with a boundary by finding a hole that is really inside candidate, calculating the rates of the width and height of face and the position of boundary. After that, we calculate the rate of eyes by width of face and then find the last eye, as shown in Fig. 6. Finally, we find a hole which is a possible mouth to make a triple hole and check whether this triple hole is a component of face or not by the first case.

3.2.3 Solving the last case

If two eyes are connected, it makes a strange size region besides the size of candidate. We choose the strange region and find the eyes by the rate of it and width of face, in Fig.7. After that, we use the first case to find a triple hole which is a component of face.

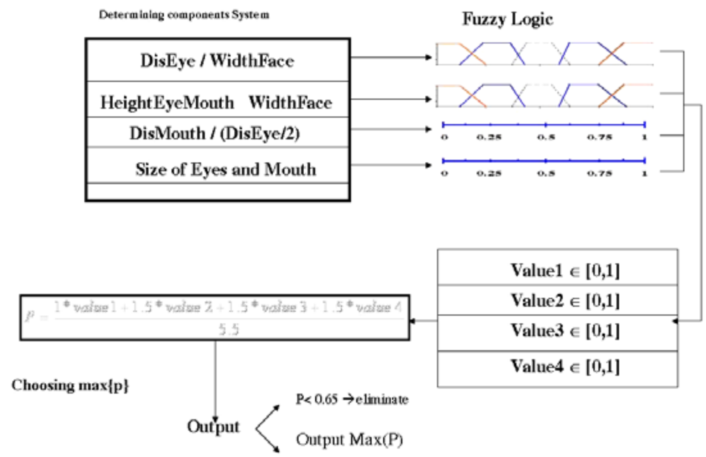

Fig. 4. Solving the first case

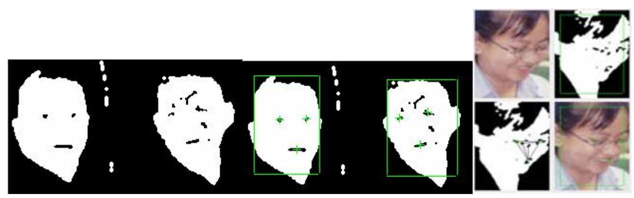
Fig. 5. Result of the first case

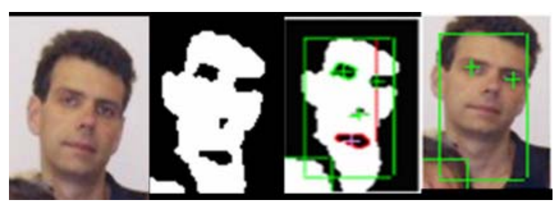
Fig. 6. Result of the second case


Fig. 7. Result of the last case

3.3 Probability of rules

When we have a set of three holes that is a possible component of face, we use the following formula:

$$p = \frac{\sum m_k w_k}{\sum m_k} * 100\% \quad (2)$$

m_k: probability of the k^{th} rule.
w_k: weight of the k^{th} rule.
p: probability of face

To find the reliability (probability) of the candidate's being face or not by rules, each rule gives a probability and weight. We have rules as follows:

1 The centre of mouth is near the axis which is between eyes.
2 Calculate the rate of distance of centre of eyes and the width of face at the position of eyes.
3 Calculate the rate of width of mouth and the width of face at the position of mouth.
4 Calculate the rate of distance from eyes to boundary and the width of eyes.
5 Calculate the rate of distance from upper boundary to eyes and the height of face.
6 Calculate the rate of distance from upper boundary to nose and the height of face.
7 Calculate the rate of distance from upper boundary to mouth and the height of face.
8 Calculate the rate of distance from eyes to nose and the height of face.
9 Calculate the rate of distance from nose to mouth and the height of face.
10 Calculate the rate of distance from nose to mouth and distance from eyes to nose.

4 Experimental Results

Our algorithm is implemented by Java™2 of Sun Microsystems using threading technology to reduce the computation time when each border is a thread. The algorithm is evaluated on several face image databases, including family, club, group, band photo collections, etc.…, only a few of which we use to construct thresholds and fuzzy membership functions. These color images have been taken under varying lighting conditions and with complex backgrounds. Furthermore, these images contain multiple faces as in Fig. 9, with variations in color, positions as shown in Fig. 8, scale, orientation, 3D pose, and facial expression in Fig. 10.

Our algorithm can detect multiple faces of different sizes with a wide range of facial variations in an image. All the algorithmic parameters in our face detector have been statistically determined over a small image database; however, the result is still good on other images. The average computation time of algorithm is relatively fast because we make the detection only on border pixels by fuzzy logic.

Using 469 images, which are approximately 786x550 pixels in size, we have evaluated 1859 real faces in total, where the total number of detected faces is 1621 and the total number of wrongly detected is 35 candidates.

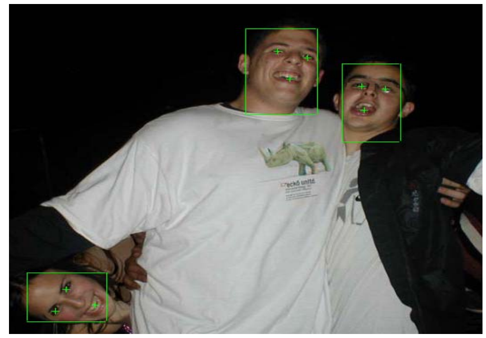

Fig. 8. A typical sample showing successful detection in a rotated image

Table 1. Comparison of running times.

Computer	RAM	Operating System	Time
PIII – 866MHz	128M	Windows XP	3.8 s
PIV – 1.7GHz	256M	Windows XP	3.1 s
PIV – 2.4GHz	512M	Windows Server 2K	1.6 s

The correct ratio is over 87%, the correct ratio of component's detection is 90% and the average separating ratio is 80%. This correct ratio is better than that in [1], but the running time is nearly the same.

5 Conclusion

Among the face detection algorithms based on skin color and facial components, we have presented a new approach working on border pixels of holes which are possible components of a face. The computation time is dramatically decreased. Moreover, our method embraces various face cases in images and gets satisfactory results. Particularly, we can separate connected faces. The separation on border pixels helps us get acquainted with new interesting algorithms such as Fast Marching. So the algorithm can be used in real-time applications.

However, the algorithm is easily influenced by a varying condition of light and the view of landscape; then the correct ratio is not over 90 percent and the failure ratio is not high. We try to build a method to integrate the proposed algorithm and the rules, and to combine with the rules of [1] and add more rules. It is needed to increase the correct ratio and to decrease the computation time concurrently.

6 References

[1] Pham The Bao, Jin Young Kim, Seung You Na, "Fast Multi-Face Detection in Color Images Using Fuzzy Logic", International symposium on Intelligent Signal Processing and Communication Systems, Hong Kong, 2005.

[2] R.-L. Hsu, M. Abdel-Mottaleb and A. K. Jain, “Face Detection in Color Images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696-706, May 2002

[3] Shixiong Li, “Multi-faces Location in Color Image”, Thesis, Concordia University, Montreal, Quebec, Canada, 2002.

[4] Dale Brisinda, “Face Detection from Constituent Features and Energy Minimizing Spring-Based Facial Templates”, Thesis, Department of computer science, The University of Calgary, 1998.

[5] Jin Ok Kim, Jin Soo Kim, Young Ro Seo, Bum Ro Lee, Chin Hyun Chung, Key Seo Lee,Wha Young Yim, and Sang Hyo Lee, “On Extraction of Facial Features from Color Images”, ICCSA 2004, LNCS 3046, pp. 964-973, Springer-Verlag Berlin Heidelberg, 2004.

[6] Saman Cooray and Noel O’Connor, “Facial feature extraction and principal component analysis for face detection in Color Images”, ICIAR 2004, LNCS 3212, pp. 741-749, Springer-Verlag Berlin Heidelberg, 2004.

[7] Kenji Iwata, Hitoshi Honggo, Kazuhiko Yamamoto, and Yoshinori Niwa, “Robust Facial Parts Detection by Using Four Directional Feature and Relaxation Matching”, The 7th International Conference on Knowledge-Based Intelligent Information & Engineering Systems, Oxford, UK, 2003.

[8] M.Castrillon Santana, J.Lorenzo Navarro, J.Cabrera Gamez , “Detection of Frontal Faces in Video Streams”, Biometric Authentication, LNCS 2359, pp. 91-102, Springer-Verlag Berlin Heidelberg, 2002.

[9] Xiangping Wang and Xingming Zhang, “Gabor Wavelet-based Eyes and Mouth Detection Algorithm”, Sinobiometrics 2004, LNCS 3338, pp. 110-117, Springer-Verlag Berlin Heidelberg, 2004.

[10] Carlos Morimoto, Dave Koons, Amon Amir, Myron Flickner, “Real-Time Detection of Eyes and Faces”, In Proc. Workshop on Perceptual User Interfaces, pages 117–120, 1998.

[11] Kyongpil Min, Junchul Chun, and Goorack Prak, “A Nonparametric Skin Color Model for Face Detection from Color Images”. PDCAT 2004, LNCS 3320, pp. 116-120, Springer-Verlag Berlin Heidelberg, 2004.

[12] C. Garcia, G. zikos, G. Tziritas, “Face Detection in Color Images using Wavelet Packet Analysis”, Multimedia Computing and Systems, 1999. IEEE International Conference on Volume 1, 7-11 June 1999 Page(s):703-708 vol.1 Digital Object Identifier 10.1109/MMCS.1999.779285.

[13] Allan Hanbury, “Mathematical Morphology from Erosion to Watersheds”, Lecture of Venia University of technology, 9-1983.

[14] Yen-Yu Lin, Tyng-Luh Liu, and Chiou-Shanm Fuh, “Fast Object Detection with Occlusions”, ECCV 2004, LNCS 3021, pp. 402-413, Springer-Verlag Berlin Heidelberg, 2004.

Fig. 9. Typical samples showing separated faces

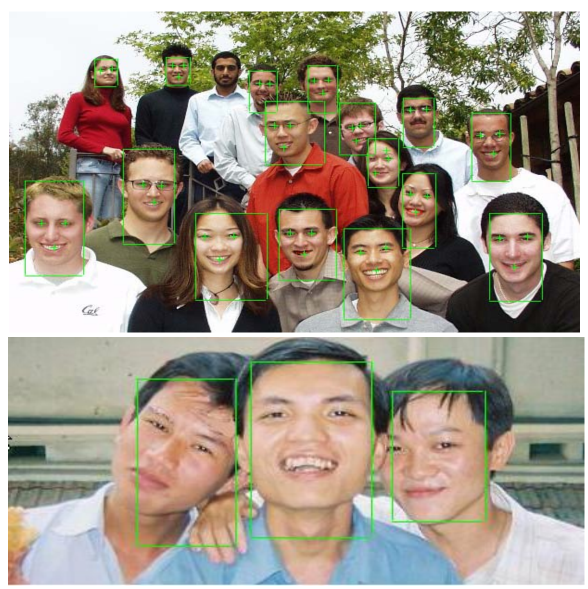

Fig. 10. The connected faces