Human Face Detection Using Geometric Triangle Relationship

Chiunhsiun Lin, Kuo-Chin Fan

Institute of Computer Science and Information Engineering,
National Central University, Chung-Li, Taiwan, 32054, R.O.C.
E-mail : kcfan@ncuee.ncu.edu.tw

Abstract

The proposed system consists of two main parts. The first part is to search the potential face regions that are gotten from the triangles based on the rules of "the combination of two eyes and one mouth". The second part of the proposed system is to perform the face verification task.

The proposed face detection system can locate multiple faces embedded in complicated backgrounds. Moreover, it is able to handle different size, different lighting condition, varying pose and expression, noise and defocus problems, and the problem of partial occlusion of mouth and sunglasses.

Experimental results demonstrate that an approximately 98% success rate is achieved and the relative false detection rate is very low.

Key Words: Face detection, triangle-based segmentation, weighting mask function

1. Introduction

Automatic recognition of human faces is one of the most difficult and important problems in the areas of pattern recognition and computer vision. As we know, a successful face detection process is the prerequisite to facilitate later face recognition task. Therefore, we should not treat face detection process merely as a preprocessing a face recognition system. On the other hand, we should deal with face detection problem as important as the face recognition problem. Abundant of researches has been conducted on human face detection. Some successful systems have been developed and reported in the literature, such as human face recognition [1,2,3,4], and human face detection [5,6,7,8,9,10,11,12].

However, most of the aforementioned reports impose many restrictions, such as they cannot allow varying pose, expression, and noise and defocus problems. In order to remove these restrictions, we develop a robust and efficient system that is suitable for real time face detection. The designed system is composed of two principal parts as shown in Fig. 1. The first part of the designed system is to search the potential face regions. The second part is to perform the task of face verification for each potential face region.

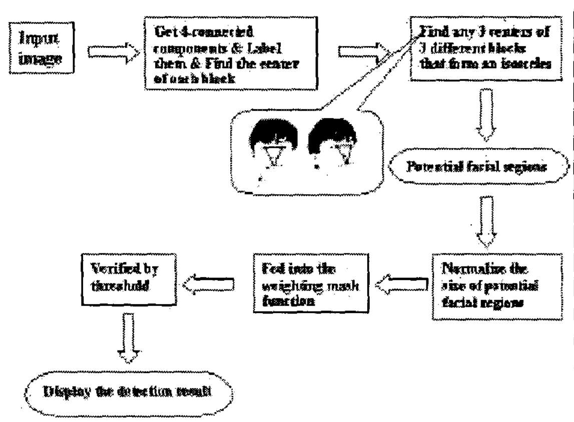


Figure 1. Overview of our system

The proposed face detection system knows how to locate multiple faces oriented in complicated background automatically. Furthermore, it can handle different size, dissimilar lighting condition, varying pose and expression, and noise and defocus problems. In addition to cope with the problem of partial occlusion of mouth and sunglasses.

The rest of the paper is organized as follows. In section 2, segmentation of the potential face regions based on some geometric rules is described. In section 3, each of the normalized potential face regions is fed to the weighting mask function to verify whether the potential face region really contains a face. Experimental results are demonstrated in section 4 to verify the validity of the proposed face detection system. Finally, conclusions are given in section 5.

2.Segmentation of potential face regions

The main purpose of this process is to find the regions in an input image that might potentially contain faces. There are four steps in the first part of the designed system. Firstly, read in an image, it could be either a color or gray level image. Then, convert this input image

0-7695-0750-6/00 $10.00 © 2000 IEEE

to a binary image. If this input image is not a grayscale image, transform the input image to grayscale format, and then convert this grayscale image to binary image by thresholding. Secondly, label all 4-connected components in the image to form blocks and find out the center of each block. Thirdly, detect any 3 centers of 3 different blocks that form an isosceles triangle Fourthly, clip the blocks that satisfy the triangle criteria as the potential face region.

2.1. Preprocess the inputted image to a binary image

In this step, we first convert the input image to a binary image. For example, when we read in a RGB color image, we will preprocess the input image to gray level image by eliminating the hue and saturation information while retaining the luminance. Then, binarize the gray level image to a "binary image" by simple global thresholding with threshold T because the objects of interest in our case are darker than the background. Before proceeding to the next step, we perform the opening operation (erosion first, then dilation) to remove noise, and then the closing operation (dilation first, then erosion) to eliminate holes. The detail of opening and closing operations can be found in [13].

2.2. Label all 4-connected components and find the center of each block

Here, we use raster scanning (left-to-right and top-to-bottom) to get 4-connected components, label them, and then find the center of each block. The detail of raster scanning can be found in [13].

2.3. Find any 3 centers of 3 different blocks that form an isosceles triangle

From careful observation, we discover that two eyes and one mouth in the frontal view will form an isosceles triangle. This is the rationale on which the finding of potential face regions is based. We could search the potential face regions that are gotten from the criteria of *"the combination of two eyes and one mouth"*.

2.3.1. The matching rules for finding an isosceles triangle

If the triangle $i\ j\ k$ is an isosceles triangle as shown in Figure 2(a), then it should possess the characteristic of "the distance of line $i\ j$ = the distance of line $j\ k$". From observation, we discover that the Euclidean distance between two eyes (line $i\ k$) is about 90% to 110% of the Euclidean distance between the center of the right/left

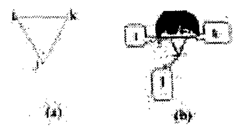


Figure 2. (a)The isosceles triangle *i j k*. (b) Three points (*i*, *j*, and *k*) form an isosceles triangle.

eye and the mouth. Due to the imaging effect and imperfect binarization result, a 25% deviation is given to absore the tolerance. The first matching rule can thereby be stated as (abs (D (i, j)-D (j,k))< 0.25* max (D(i, j), D(j, k))), and the second matching rule is (abs(D(i, j)-D(i, k)) < 0.25*max(D(i, j), D(j, k))). Since the labeling process is operated from left to right then from top to bottom, we can get the third matching rule as "$i < j < k$". Here, "abs" means the absolute value, "D (i, j)" denotes the Euclidean distance between the centers of block i (right eye) and block j (mouth), "D (j,k)" denotes the Euclidean distance between the center of block k (left eye) and block j (mouth), "D (i,k)" represents the Euclidean distance between the centers of block i (right eye) and block k (left eye). For example, as shown in Figure 2(b), if three points (i,j, and k) satisfy the matching rules, then we think that they form an isosceles triangle.

After we have found the isosceles triangle, it is easy to get the coordinates of the four corner points that form the potential facial region. Since we think the real facial region should cover the eyebrows, two eyes, mouth and some area below mouth [9], the coordinates can be calculated as follows. Assume that (Xi, Yi), (Xj, Yj) and (Xk, Yk) are the three center points of blocks i, j, and k that form an isosceles triangle. (X1, Y1), (X2, Y2), (X3, Y3), and (X4, Y4) are the four corner points of the face region as shown in Figure 3. X1 and X4 locate at the same coordinate of (Xi - 1/3*D (i, k)); X2 and X3 locate at the same coordinate of (Xk + 1/3*D (i, k)); Y1 and Y2 locate at the same coordinate of (Yi + 1/3*D (i, k)); Y3 and Y4 locate at the same coordinate of (Yj - 1/3*D (i, k)); where D (i, k) is the Euclidean distance between the centers of block i (right eye) and block k (left eye).

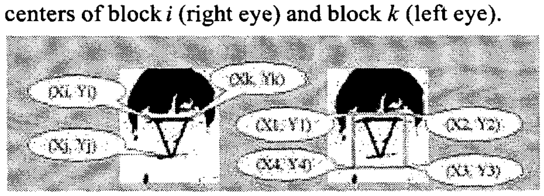


Figure 3. Assume that (X*i*, Y*i*), (X*j*, Y*j*) and (X*k*, Y*k*) are the three center points of blocks *i*, *j*, and *k*, respectively. The four corner points of the face region will be (X1, Y1), (X2, Y2), (X3, Y3), and (X4, Y4).

$$X1 = X4 = Xi - 1/3 * D(i, k); \quad (1)$$
$$X2 = X3 = Xk + 1/3 * D(i, k); \quad (2)$$
$$Y1 = Y2 = Yi + 1/3 * D(i, k); \quad (3)$$
$$Y3 = Y4 = Yj - 1/3 * D(i, k); \quad (4)$$

3.Face verification

The second part of the designed system is to perform the task of face verification. In the previous section, we have selected a set of potential face regions in an image. In this section, we propose an efficient weighting mask function that is applied to decide whether a potential face region contains a face. There are three steps in this part. The first step is to normalize the size of all potential facial regions. The second step is to feed every normalized potential facial region into the weighting mask function and calculate the weight. The third step is to perform the verification task by thresholding the weight obtained in the precious step.

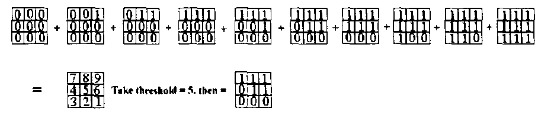
3.1.Normalization of potential facial regions

Normalization of a potential face region can reduce the effects of variation in the distance and location. Since all potential faces will be normalized to a standard size (e.g. 60 * 60 pixels) in this step, the potential face regions that we have selected in the previous section are allowed to have different sizes. Here we resize the potential facial region using "bicubic" interpolation technique.

3.2.Weighting mask function and weight calculation

If the normalized potential facial region is really contains a face, it should have high similarity to the mask that is formed by 10 binary training faces. The idea is similar to neural network, but does not need so many training samples. Every normalized potential facial region is fed into the weighting mask function that is used to compute the similarity between the normalized potential facial region and the mask. The computed value can be utilized in deciding whether a potential region contains a face or not.

The method for generating a mask is to read in 10 binary training masks that are cut manually from the facial regions of images, then add the corresponding entries in the 10 training masks to form an added mask. Next, binarize the added mask by thresholding each entry. For example, we have 10 masks and the size of each mask is 3*3. The first mask is formed by 9 "zero" ("zero" represents white pixel), so the first mask is a white 3*3 block. The 10th mask is produced by 9 "one" ("one" represents black pixel), so the 10th mask is a black 3*3 block. We added these 10 masks together and get an added mask with the value of each entry as [7 8 9 4 5 6 3 2 1]. If we select the threshold value 5 (if the value of each entry is larger than 4, then we assign its value as 1; otherwise we assign its value as 0.), we get the final mask that has the values of [1 1 1 0 1 1 0 0 0]. The illustrating procedure is depicted as shown below:



For all pixels of the potential facial region and the mask: If both of the potential facial region and the mask contain those pixels at the same location of the parts of two eyes, the nose, and the mouth (both of them are black pixels.), then weight = weight + 6. If both of the potential facial region and the mask contain those pixels at the same location of the parts of the skin of the face (both of them are white pixels.), then weight = weight + 2. If the pixel of potential facial region is black and the pixel of the mask is white, then weight = weight - 4. If the pixel of potential facial region is white and the pixel of the mask is black, then weight = weight −2. The experimental results show that the values of + 6, + 2, - 4, and − 2 as given above can obtain the best match of the facial region. In other words, the range of threshold values is the narrowest.

3.3.Verification

After we have calculated the weight of each potential facial region, then a threshold value is given for decision making. Once a face region has been confirmed, the last step is to eliminate those regions that overlap with the chosen face region, and then exhibit the result.

Verification of the frontal view - A set of experimental results demonstrates that the threshold values of the frontal view should be set between 4,000 and 5,500.

4.Experimental results and discussion

In this section, a set of experimental results is demonstrated to verify the effectiveness and efficiency of the proposed system. There are 500 test images (include 450 different persons) containing totally 600 faces which are used to verify the validity of our system. Some test images are taken from digital camera, some from scanner, and some from videotape. The sizes of the test images range from 10*10 to 640*480 pixels (actually our system could handle any size of image.) In these test images, human faces were presented in various environments.

The execution time required to locate the precise locations of the faces in the test image set is dependent upon the size, resolution, and complexity of images. For example, the gray level image which is 200*307 pixels as shown in Figure 4(a) need less than 2.5 seconds to locate the correct face position using PII 233 PC. The gray level image which is 200*145 pixels as shown in Figure 4(b) need about 28 seconds to locate the correct face position because the background is more complex than Figure 4(a). Figure 4: Experimental results of gray level/color images with simple/complex backgrounds.

Figure 4: Experimental results of gray level/color images with simple/complex backgrounds.

5.Conclusions

In this paper, a robust and effective face identification system is presented to extract face in various kinds of face images. Moreover, it can manage distinct size, changed lighting condition, varying pose and expression, and the noise and defocus problem. In addition, the presented system can also cope with the problem of partial occlusion of mouth and wearing sunglasses. The experimental results reveal that the proposed method is better than traditional methods in terms of altered circumstance, efficiency, and accuracy.

6.References

[1] M. Turk and A Pentland, "Eigenfaces for recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71--86, 1991.

[2] David J. Beymer, "Face recognition under varying pose", Technical Report, MIT AI Lab, AI memo 1461, December 1993.

[3] D. Valentin, H. Abdi, A. O'toole, and G. Cottrell, "Connectionist models of face processing: A survey", Pattern Recognition, vol. 27, no. 9, pp. 1209-1230, 1994.

[4] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition", in IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington, Jun. 1994, pp. 84-91.

[5] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection", in Proc. Image Understanding Workshop, Monterey, Calif., Nov. 1994, pp. 843-850.

[6] T. K. Leung, M. C. Burl, and P. Perona, "Finding faces in clustered scenes using random labeled graph matching", in Proc. Computer Vision and Pattern Recognition, Cambridge, Mass., Jun. 1995, pp. 637-644.

[7] H. A. Rowley, S. Baluja, and T. Kanade, "Human face detection in visual scenes", Tech. Rep. CMU-CS-95-158R, Carnegie Mellon University, 1995. (The paper is gotten from http://www.cmu.edu/~har/faces.html)

[8] S. Y. Lee, Y. K. Ham, and R. H Park, "Recognition of human front faces using knowledge-based feature extraction and neuro-fuzzy algorithm", Pattern Recognition, vol. 29, no. 11, pp. 1863-1876, 1996.

[9] P. Juell and R. Marsh, "A hierarchical neural network for human face detection", Pattern Recognition, vol. 29, no. 5, pp. 781-787, 1996.

[10] K. Sobottka and I. Pitas, "Extraction of facial regions and features using color and shape information", in Proc. 13th International Conference on Pattern Recognition, Vienna, Austria, Aug. 1996, pp. 421-425.

[11] H. Wu, Q. Chen, and M. Yachida, "A fuzzy-theory-based face detector", in Proc. 13th International Conference on Pattern Recognition, Vienna, Austria, Aug. 1996,

[12] C. C. Han, H. Y. Mark Liao, G. J. Yu, and L. H. Chen, "Fast face detection via morphology-based pre-processing", in Proc. 9th International Conference on Image

[13] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", copyright © 1992 by Addison-Wesley Publishing Company, Inc.