A Frontal Face Detection Algorithm Using Fuzzy Classifier

Ramazan Tasaltin and **Osman Taylan**

Qassim University, Electrical Engineering Department, Qassim, and King Abdulaziz University, College of Engineering, Industrial Engineering Department, P.O box 80204, Jeddah 21589, Saudi Arabia
otaylan@kau.edu.sa

Abstract. This paper describes a new approach for detecting faces whose size, position and pose are unknown in an image. Skin color and face border information are the primary key parameters to develop a fuzzy rule based classifier and to extract a face candidate from an image. At first, face candidate blobs were extracted using skin color information, then the blob was divided into four regions and the face borders are approximated by polynomials. Second, reference polynomials were extracted by examining different faces. As a result, the deviations from reference polynomials were used by a fuzzy classifier to decide whether the blob contains a face or not.

Keywords: Face detection, Face border information, Fuzzy inference system, fuzzy classifier.

1. Introduction

Detecting human face automatically in a cluttered image is an important step to a fully automatic face recognition system. Human face detection is difficult because there can be huge variations in the appearance of face patterns and many of these variations are difficult to parameterize. The starting problem in most of the cases of facial feature extraction is face detection, which deals with locating faces in images. Face detection is a problem on its own, on which significant research has been conducted. Different approaches have been applied for face detection. More than 150 approaches have been reported [1]. All methods are classified under the following groups. The first of which are knowledge-based methods that encode human knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features

[2]. Second, feature invariant approaches: These algorithms aim to find structural facial features, grouping of edges [3, 4], texture space, gray level dependence matrix of face pattern, skin color [5], and face shape [6]. Face detection tasks are accomplished by manipulating distance, angles, and area measurements of the visual features derived from the scene [7, 8]. The third method entails template matching such that several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the stored patterns are computed for detection. Predefined face templates [9], deformable templates are some examples for this method. And finally, in appearance-based methods; models (or templates) are learned from a set of training images and are used for detection. These methods include Eigen face [10], Distribution based [11], Neural network [12], Support vector machine [13], Naive Bayes classifier [14], Hidden Markov model [15], Information theoretical approach [16, 17], and Wavelet analysis.

The first part of this study is to detect a face-like blob. This is accomplished by using YCbCr color space model. YCbCr color space has the property that almost all face colors are condensed in a narrow region of Cb-Cr plane [6]. Our experiments have shown that a single face can be extracted from an image that contains multi-faces and complex background by properly selecting the Cb and Cr interval.

It is assumed that the face is viewed from the front. However, the algorithm developed is able to produce almost the same results when the angle of view is varied 35^o towards left or right and 20^o towards the front and the back. The algorithms should be modified if the view angle is more than these limits, then face-like blobs are extracted from an image. Each blob is treated separately. One blob is divided into four sections. The outer edges of the sections are approximated by polynomials. By examining more than 500 faces, we have calculated average reference polynomials for outer edges of the faces. A polynomial that is obtained from an image is compared to the reference polynomial. The deviations from reference polynomials are used as a measure to determine whether the blob belongs to a face or not. Because of high complexity of face recognition problem and due to large variability of conditions to capture the images and diversity of human faces, Fuzzy Inference System (FIS) is used. FIS classifies face and non-face blobs using the deviations between the reference polynomial and the polynomial that is obtained from an image. Figure 6 illustrates a polynomial fitted to the face border on the image which clearly shows the deviations. Similarly, the comparison of face parts polynomial with reference polynomial is given in Fig. 7. The deviations between these polynomials are measured and the polynomial error is identified by fuzzy linguistic values. The fuzzy linguistic set used to define the deviation is as follows: *(face, rather face, low probability face, not face).* Fuzzy clustering algorithm was used to decide the number of rules for the FIS. In this study, fuzzy classifier works as a FIS to decide whether the blob contains a face or not. FIS is a computing framework based on the concepts of fuzzy sets, fuzzy IF-THEN rules, and fuzzy reasoning. In this study, a multi-input single output (MISO) FIS was developed which yielded significant results. The face and non-face blobs are determined by defuzified results of FIS mapping from fuzzy inputs over output.

The last part of the paper is devoted to face verification. If the outcome of the fuzzy classifying algorithm is that the blob contains a face, then the last part is executed. The last part assumes that the blob contains a face and face borders are known. The algorithm tries to find two eyes or one eye in case that the face is rotated towards left or towards right.

2. Utilization of Skin Colors

2.1 Color Space Selection

Many color spaces (HSI, YCbCr, YES, RGB, HSV) have been used in segmenting skin regions in color images. Literature survey [1] shows that the YCbCr color space is one of the successful color spaces in segmenting the skin color accurately, mainly because the chrominance components are almost independent of luminance component in the space. Although skin color varies from person to person, it tends to get clustered into a compact region in CbCr space. We have also tested RGB, HSV, HSU color spaces. It has been found that Ycbcr color space is the most suitable way to distinguish face and non-face regions. Because all the human face colors are transformed into a very narrow region of Ycbcr color space. Since we use only Cb and Cr components of Ycbcr color space, computational burdens are also reduced dramatically in Ycbcr color space.

2.2 Face Skin Colors

A row image is projected into a low dimensional subspace by simply linearly compressing the RGB image. The size of the compressed images is typically 400x600 pixel. If the original image is less than 400x600 pixel then the compression is not performed.

The new compressed RGB image is projected into CbCr space. Although skin colors are compact in CbCr color space, it still has small variances from person to person. If the background color is similar to face-color, then we will have an additional problem for accurate detection of face-like regions. To overcome this problem we try to capture the image region that is within certain values of Cb and Cr, *i.e.*, try to detect the region that satisfies the condition ($Cb_{min}<Cb<Cb_{max}$ and $Cr_{min}<Cr<Cr_{max}$). These threshold (Cb_{min}, Cb_{max}, Cr_{min},

Cr_{max}) values are sweeped through covering the whole possible face-CbCr values. Face-like regions are detected perfectly, if the correct threshold values are selected. Figure 1 shows a typical example. Those pixels that lie inside the threshold values ($Cb_{min}<Cb<Cb_{max}$ and $Cr_{min}<Cr<Cr_{max}$) are set to 1 (white) and all other pixels, that are outside of this region are set to zero (black).

When we select Cb_{max}=160, Cb_{min}=135, Cr_{max}=125, Cr_{min} =100 as in Fig. 1(b), all six faces become clear. However, if Cb_{max}=160, Cb_{min}=145, Cr_{max}=110, Cr_{min} =90 is selected as in Fig. 1(c), then only four faces are selected. Similarly only two faces became clear in Fig. 1(d) for the threshold values that are shown on top of the Fig. 1. Thus, appropriately selecting Cb_{max}, Cb_{min}, Cr_{max}, Cr_{min} values, it is possible to identify location of faces that have different skin colors and have different illuminations.

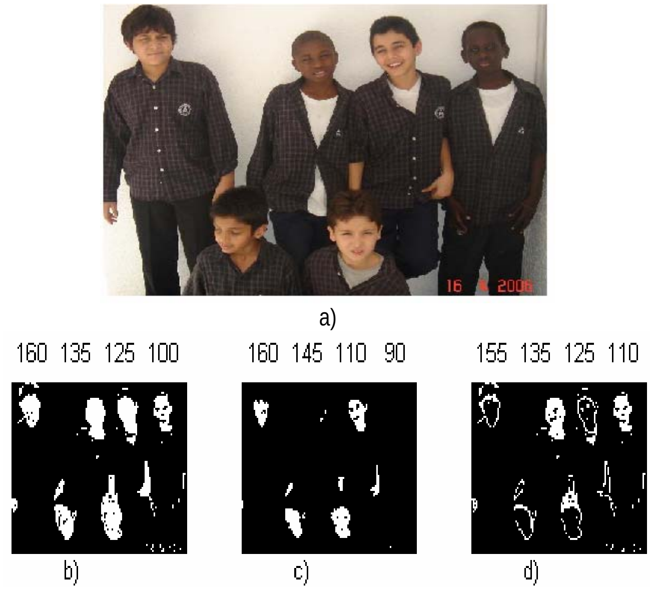

Fig. 1. Original picture and its projection using YCbCr color space: a)Original picture. (b),(c),(d) projection of the picture using different Cb and Cr intervals. The numbers on top of the sub Figs denote Cb_{max},Cb_{min}, Cr_{max},Cr_{min} values.

Pietrowcew [18], in his proposed algorithm, models the shape of a human face as ellipse with constrained proportions, similar to Seguier *et al*. In this approach, for each dimension of ellipse axis, one accumulator describing all possible ellipses in the image with constrained size is defined. Because human heads are rather longer than wider in general, hence, he limits the proportions of the ellipses axes a_i and b_i in the considered accumulators. In reality, shape of human face does not respond in an image to ellipse too precisely. He applies weighted fuzzy Hough transform to the parameter space which is directly proportional to the gradient amplitude at a given edge point to propagate the estimated weight to neighbouring ellipses. Hough transform is a tool allowing for localization of specific shapes in images, on the basis of objects contours present in them. Shape detection is carried out by analysis of edge points found in the image. In order to closely approximate face oval, it is necessary to add additional parameterization during Hough transform calculation. In same situations, Hough transform mistakenly generate ellipses. To solve this particular problem, each cell in Hough transform parameter space is split into four sub-cells, corresponding to ellipses quarters. Values in Hough transform parameter space are calculated only for edge pixels, in order to get them the luminance component of image is blurred (fuzzified) by Gaussian filter.

3. Face-Like Blob Detection

When we set a face region white, the next problem is to find in which parts of the image the face-like blobs exist. In order to solve this problem, the total image was divided into small rectangles. If a rectangle contains more white pixels than a threshold value, then this rectangle is considered as part of a possible face region. The size of the rectangle is very small, typically less then 0.5% of the total image size. Those small neighbor rectangles are combined together to constitute a big blob. A simple algorithm is used to remove the small dots around each big blob. The algorithm checks the location and the white/black ratio of a rectangle. If the location is on the outer part of the big blob and the white/black ratio is less than a threshold value then that rectangle is considered as noise. The remaining part is a face candidate. Figure 2 shows a typical result. Image of Fig.1 is processed using the outlined method and face-like blobs are subtracted from the Ybcbr color space image.

Fig. 2. Face-like blobs obtained from the image of Fig. 1(a).

Now, the main question is “Is each block a face or not”. This is accomplished by an iterative algorithm that will be explained in sections 4,5,6. The result of this algorithm will be that each of these blobs is a face or not a face. If it is found that a blob is a face, then the smallest rectangle that contains the face are subtracted from the row image. Figure 3 shows a typical result.

3.1 Existence of Ear(s) in an Image

In most cases one ear or both ears are hidden in a face image. This may be due to a rotation of the face towards left or right, or ear(s) may be covered by the hair. We detected the ear by measuring the distance between the top-horizontal axis and the first white pixels in each column as shown in Fig. 4. The ear exhibits itself as valley followed by a peak.

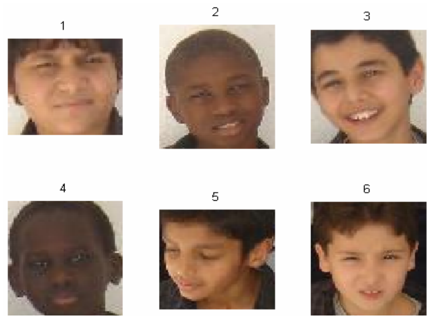

Fig. 3. Faces captured from the image of Fig. 1(a).

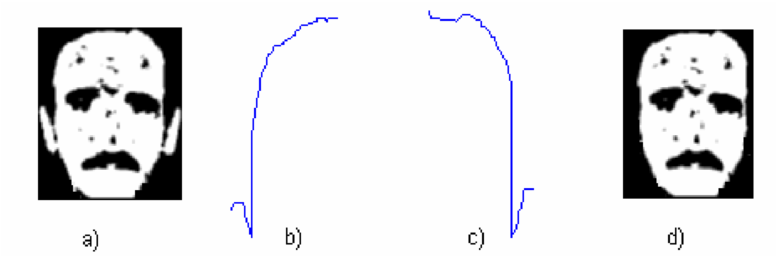

Fig. 4: (a) A sample head image in CbCr domain, (b) The sketch of top-to-image vertical distance (left side), (c) The sketch of top-to-image vertical distance (right side), (d) remaining head image.

3.2 Removing the Bottom of Neck Region

If a face-candidate blob includes a bottom side of neck region, then bottom of neck part is removed. Neck is detected using a similar method as in the case of ear detection. If we plot the horizontal distance between the Y axis and the image, then the neck exhibits itself as a valley.

3.3 Polynomial Approximations for the Face Borders

There are many face templates in the literature that approximate face region shape. They use certain types of templates that a face region fits exactly. However the exact face region cannot be detected using image-YCbCr domain data. Hair might cover the upper part of the face region. As it appears in Fig. 5(a) in some cases face region is displayed in a mess. Thus, there is a little gain for using exact face template in the first place. Instead of using the total face region, the face region may be divided into 4 parts as shown in Fig. 5 (b).

The outer borders of the four regions can be used as face templates. We have used polynomial approximations for the four different outer parts as shown in Fig. 6 and 7. When the face is rotated towards left or right, these polynomials change slightly. If the rotation is small then the deviation can be ignored. If an ear is detected in an image, then the ear is removed for polynomial approximation, because ear in an image produces irrelevant polynomial coefficients. In this approach, "What should be the order of the polynomial," is a vital question that must be studied. Second and third and higher order polynomials were tried, and it is found that that second order polynomials are the most appropriate. This is because each face has a different view angle and the best approximation is the rough approximation.

a) b)

Fig. 5: a) Face image in YCbCr domain, b)Four face parts.

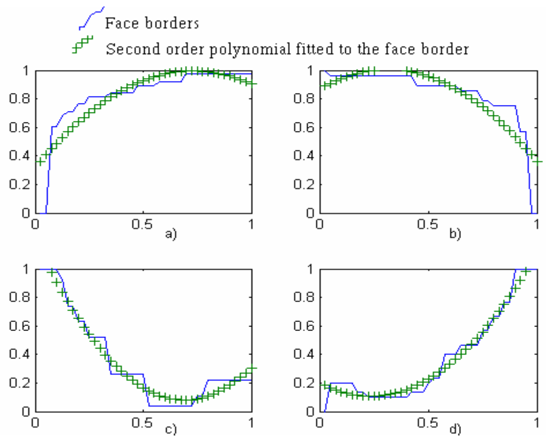

Fig. 6. Polynomial fitted for four parts of faces.

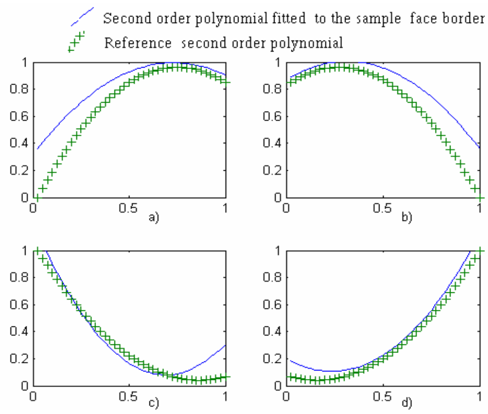

Fig. 7. Comparing Polynomials of four face parts with reference polynomials.

4. Face Detection Algorithm

The developed algorithm is summarized as follows:

- Input: RGB image data
- Convert RGB image into YCbCr space model
- Select an upper and lower bounds for Cr and Cb values. Pixels that lie between the upper and lower bounds are set to white (255), and pixels that are outside the bounds are set to black (0).
- Detect possible face-like white clusters rectangles using the method described in Section 3.
- Select one rectangle and apply fuzzy logic modeling approach to decide if the rectangle contains a face or not.
- If all rectangles are processed, then go to step 4, otherwise go to step 5.
- If all possible Cr and Cb intervals are tried then stop the algorithm, otherwise go to step 3.

5. How to Describe a Face?

Previous sections are concerned with acquiring a rectangle that contains many white pixels. The question now is "does this rectangle contains a single face?" We assume that the face is in frontal position, and may have small rotations (towards left or right and towards up or down). We are using the following criteria:

- Existence of ears (2)
- The areas between the approximated polynomial and the reference polynomial (4)
- The variation of the real face borders around the approximated polynomial (4).

Numbers in brackets denotes the number of parameters. Thus a total of 2+4+4=10 input parameters are used. Using these 10 parameters, we must decide whether the blob has a face or not. Fuzzy Inference System (FIS) is required in this part to decide whether the blob contains a face or not.

5.1. Fuzzy Inference System for Face Detection

Fuzzy inference system (FIS) is a fuzzy modeling approach and a computing framework based on the concepts of fuzzy rule base which contains the selection of fuzzy rules, and membership functions, and the reasoning mechanism. FISs have been successfully used in a wide variety of applications [19]. In this study, the fundamental question is treated mathematically for the upper and lower bounds for Cr and Cb values. Pixels that lie between the upper and lower bounds are set to white (255), and pixels that are outside the bounds are set to black (0) in order to detect possible face-like white clusters rectangles. Fuzzy logic modeling approach was used to decide that the rectangles outcome contains a face or not. The FIS in this study is a multi-input single output (MISO) fuzzy system. $f : R^n \to R$. The n-array $X = (X_1, X_2, ..., X_n)^T$ are denoting the input variables. X_1, X_2, X_3, X_4**:** are the areas between the real face borders and the approximated polynomial. Hence, X_1 is the left top part of face, X_2 is the right-top, X_3 is the bottom left part, and X_4 is the bottom right parts of the face.

Similarly, X_5, X_6, X_7 *and* X_8 are the areas between the approximated polynomial and the reference polynomials. Hence, X_5 is the left top part of face, X_6 is the right top part, X_7 is the bottom left, and X_8 is the bottom-right parts respectively. X_9, X_{10} are parameters that denote probability of existence of right and left ears, respectively.

T(x) is the fuzzy linguistic terms of x which is the set of linguistic values of A_i^j .

$$A_i^j = \{\textit{Face, Rather Face, Low\_P\_Face, Not Face}\}.$$

Similarly, the terms set of output variable y ($Decision$) given in Fig. 8 is also,

$$B^j = \left\{ \textit{Face, Rather\_Face, Low\_P\_Face, Not Face} \right\}$$

For each linguistic variable X_i, U_i is the universe of discourse or collection of possible patterns $X_i \in U_i, i = 1,2,....,n.$ and V is the universe of discourse of output, y. There are $m = 4$ membership functions for each linguistic input variable $X_i, i = 1,2,.....,n$ and output variable, y. In order to produce membership measurements of each variable with respect to fuzzy sets A_i^j and B^j, respectively, Eq. (1) is used, that is:

$$\mu_{A_i}^j(x_i):U_i \to [0,1],\ \mu_{B^j}(y):V \to [0,1],\quad j=1,2,....,m \tag{1}$$

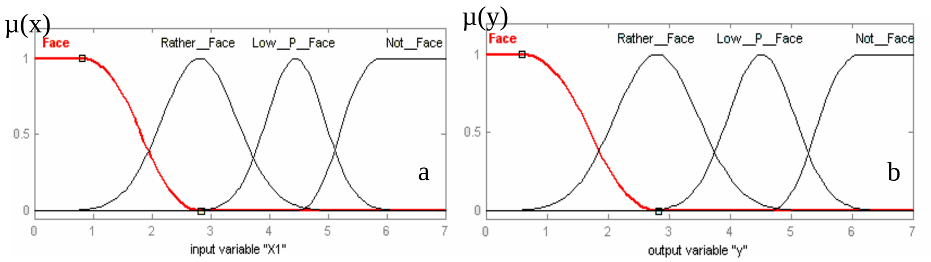

Fig. 8. Fuzzy membership functions and their term sets for input variable X1 (a), and for output variable y (b).

In fuzzy modeling, it is important to determine reasonable membership functions to maintain appropriate linguistic meanings. Nonlinear, bell shaped membership functions (MFs) are used to characterize the linguistic variables. The MFs map each element of input variable *(X)* to a membership grade (degree) between 0 and 1. Fuzzy relations include areas such as fuzzy control and decision making. Fuzzy relations in different product spaces can be combined through a composition operation. There are different composition operations for fuzzy relations, the best known is the max-min composition which is used in this study. The fuzzy relations matrix is as follows in Eq.(2).

$$R^j = (A_1^j\ AND\ A_2^j\ AND............AND\ A_n^j) \to B^j\ ,$$

$$\mu_{R^j}(x_1, x_2,......., x_n, y) = \mu_{A_1^j}(x_1).....\mu_{A_n^j}(x_n).\mu_{B^j}(y) \tag{2}$$

The fuzzy relations $R^j,\ j=1,2,.....,m,$ encoding the fuzzy logic rules can be aggregated to form the overall relation R by interpreting ELSE as fuzzy union in the Eq.(3), that is,

$$\mu_R(x_1,......,x_n,y) = \bigvee_{j=1}^{m} \mu_{R^j}(x_1,....,x_n,y) \tag{3}$$

Where $\vee$ denotes the pairwise max operator. If the input variable X_i takes the fuzzy sets $A_i^{'}, i=1,2,.....,n$, then the output fuzzy set or consequent $B^{'}$ can be deduced using the operation of fuzzy composition as follows:

$$B^{'} = (A_1^{'}\ AND\ A_2^{'}\ AND.......AND\ A_n^{'}) \circ R$$

where $\circ$ denotes the max-min compositional rule of inference. Thus the membership function of the consequent $B^{'}$ is

$$\mu_{B'}(y) = \underset{x_1,\dots,x_n}{\vee}\left[\left(\prod_{i=1}^{n}\mu'_{A_i}(x_i)\right).\left(\underset{j=1}{\overset{m}{\vee}}\left(\prod_{i=1}^{n}\mu_{A_i^j}(x_i)\right).\mu_{B^j}(y)\right)\right] \qquad (4)$$

Equation (4) defines a fuzzy mapping, $F(A_1', A_2', \ldots\ldots, A_n') = B'$. $\prod$ is the min operator. In practice, especially in control applications, the input fuzzy set A_i' equals numerical data a_i' [20]. Figure 9 is an illustration of how a four rule Mamdani fuzzy inference system derives the overall crisp outputs when subject to ten crisp input variables and one output variable. FIS takes crisp inputs, but the outputs it produces are almost always fuzzy sets. It is necessary to have crisp outputs to be understood by the analyzers, because FIS is a controller and the results it produces are important for checking the performance of model, and decision making, hence, the results are needed to be defuzzified. Defuzzification refers to the way a crisp value is extracted from a fuzzy set as a representative value for face detection. In general, there are five methods for defuzzifying a fuzzy set A of a universe of discourse. Centroid of area z_{COA} method was selected as a defuzzification block which is the most common one. The FIS for face detection and the defuzzifican block is shown in Fig. 9.

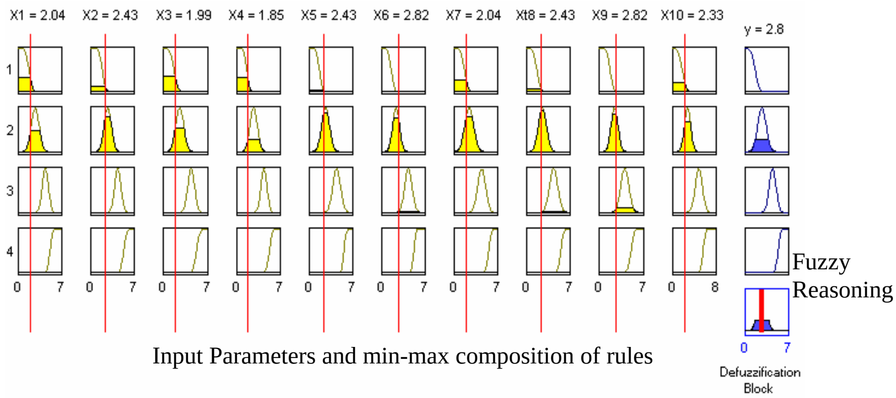

Fig. 9. The Mamdani fuzzy inference system using min-max composition for reasoning of face detection.

In this defuzzification strategy, membership value $\mu_{B'}$ is the aggregated output of Membership Function. This is the most widely adopted defuzzification strategy. The final crisp (numerical) outputs which are representative for deciding whether the blob contains a face or not for certain input values are presented in Table 1.

Table 1. Sample crisp input values versus crisp outputs and defuzzified results.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Output	Decision
1,61	1,45	1,81	1,52	1,11	1,75	2,01	3,5	1,57	1,45	1,42	Face
2,13	2,14	2,36	1,11	1,45	2,19	1,49	1,96	0,93	2,3	1,67	Face
2,19	2,48	1,96	2,3	2,13	1,96	1,96	1,45	2,48	2,24	2,69	R_Face
3,59	3,5	3,67	3,5	3,67	3,16	3,33	2,99	2,98	3,16	3,12	R_Face
3,76	4,01	3,67	4,18	4,18	3,5	4,01	4,35	3,84	3,8	4,22	Low_P_Face
4,99	5,12	4,87	5,21	4,35	4,87	5,12	5,55	4,87	5,37	4,52	Low_P_Face
5,34	5,21	5,38	5,55	5,72	5,89	5,38	5,72	5,38	5,76	5,72	Not Face
5,86	5,72	6,4	5,72	6,06	6,57	6,23	6,57	6,4	6,34	6,14	Not Face

6. Face Verification

Having decided that a blob contains a face, our next task is face verification. We verify a face by checking the following parameters:

- The height/width ratio of the face
- Existence of eyes
- Existence of the mouth

There is a significant color distance between the eye region and the skin color. Under the YCbCr color space, we can observe that the iris has a lower gray-level intensity, a higher Cb value, and a lower Cr value than the surrounding skin color. We searched eyes in 1/3 of upper part of the image. We search for symmetric eyes first, if not found, we search two eyes on approximately the same horizontal line, if this is not found, we search for one eye only. The mouth is searched as horizontal lines. We have searched eyes and mouth by using a Sobel edge detector of gray level image.

7. Increasing the Speed of the Algorithm

The speed of the algorithm depends on the background complexity and clarity of face regions. The algorithm is run for 10 set of Cb_{max}, Cb_{min}, Cr_{max}, Cr_{min} values initially. Assuming each picture has 600KB size, each run takes 0.11 second in a 3.2GHz computer. If a face (or faces) is found in a trial, the face regions are marked, so that it will not be processed for the second trial. Similarly, those regions that are a face candidate (green regions for example) are also marked so that it will not be processed for the second trial. Having completed 10 trials, if all the image regions are marked as either face or non-face regions, then the algorithm stops at this point which is taking 1.1 second for the total run. However this is a rare case. For practical situations, we need to run the algorithm for 50 (in some cases 100) set of Cb_{max}, Cb_{min}, Cr_{max}, Cr_{min} values in order to mark the image regions correctly as face or non-face areas, taking 110 seconds (for 100 run). Obviously, if we are examining pictures where camera position is constant, then we search for a very narrow region reducing the time to less then one second.

8. Experimental Results and Discussions

We have tested the algorithm given in Section 4 using different images. Accuracy of the results depends mainly on the size of the faces. Faces that occupy more than 1/10 *th* of the whole image size are detected easily. We have obtained promising results. Some of the results are displayed in Fig. 11 and 12. One face in Fig. 10 was not captured due to the face rotation. The algorithm fails to detect faces that are rotated towards left or right more than 30^o. If the rotation is known before the face detection procedure, then the algorithm can still work by using one side of the face. However we do not have any priori knowledge. Another failure is due to occlusion, *i.e*., if two or more faces are displayed one over another in the image, then it becomes impossible to identify a face in a blob. The algorithm fails if there is a long beard that covers the whole bottom side of the face region.

Fig. 10. A sample image.

Fig. 11. A typical CbCr projection of the image of Fig. 10.

Fig. 12. Faces captured from the image of Fig. 10.

9. Conclusion

This paper describes an iterative procedure for a face detection system that extracts faces in images. There are three main contributions emanating from this work. First, we explore the use of skin color and skin border information. Second, we show that fuzzy set theory can be a powerful tool for image processing and decision tool. Finally, we developed a mathematics-based face detection algorithm. Using the fuzzy set theory and inference system, we have achieved a promising face detection system that can recognize human faces successfully from various images. In this study, FIS represents the dynamic characteristics of a process by a set of fuzzy implication and variables. The fuzzy variables and their term set were established for the deviation of polynomials. A polynomial fitted to the face parts on the image and the comparison of this polynomial with reference polynomial was fitted to certain crisp intervals between (0-7). The deviations between these two polynomials were measured by fuzzy membership degrees. For example, if the output is between (0-1.8) the blob is a *'face'*, if the output is between (1.8-3.7) the blob is a *'rather face'*, if the output is between (3.5-5.2) the blob is a *'low probability face'*, if the output is greater than 3.76 then the blob is *'not face*'. The output of fuzzy model is defuzzified and the crisp values are expressible by their linguistic equivalences and are given in the last column (Decision) of Table 1. Similarly, for example, the output 1.2 ($\mu(1.2) = 0.9$) means that this output belongs to the fuzzy set *'face*' with a degree of membership 0.9 (Fig. 8 (b)), hence the blob is a *face.* The output 5.5 ($\mu(5.5) = (0.25, \;\; 0.75)$) is taken to mean that this output belongs to the fuzzy set '*low probability face'* with a degree of 0.25 and belongs to the '*not face'* set with a degree of 0.75, hence, it is not a face. The future plan is a study for deciding whether a face on an image belongs to a specific person or not.

Acknowledgement

The authors would like to express their special thanks and appreciation to Prof. Dr. Talip Alp for his valuable contribution on language of manuscript and Prof. Dr. Mehmet Akyurt for his comments and encouragement during preparation of this manuscript.

9. References

[1] **Yang, Ming-Hsuan, J. K., David** and **Ahuja, N.,** "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(1) January (2002).

[2] **Yang, G.** and **Huang, T. S.,** "Human Face Detection in Complex Background", *Pattern Recognition*, **27**(1): 53-63 (1994).

[3] **Leung, T.K., Burl, M.C.** and **Perona, P.,** "Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching", *Proc. Fifth IEEE Int'l Conf. Computer Vision*, PP: 637-644 (1995).

[4] **Yow, K.C.** and **Cipolla, R**., "Feature-Based Human Face Detection," *Image and Vision Computing*, **15**(9): 713-735 (1997).

[5] **Yang, J.** and **Waibel, A.,** "A Real-Time Face Tracker," *Proc. Third Workshop Applications of Computer Vision*, PP : 142-147 (1996).

[6] **Kjeldsen, R.** and **Kender, J.**, "Finding Skin in Color Images," *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition*, PP: 312-317 (1996).

[7] **Brunelli, R.** and **Poggio, T.,** Face recognition: Feature versus templates, *IEEE Trans. Pattern Anal. Mach. Intell.* **15**: 1042–1052 (1993).

[8] **Valentin, D., Abdi, H., O'Toole, A. J.** and **Cottrell, G.**, Connectionist models of face processing: A survey, *Pattern Recog.* , **27**: 1209–1230 (1994).

[9] **Craw, I., Tock, D.** and **Bennett, A.**, "Finding Face Features" , *Proc. Second European Conf. Computer Vision*, PP : 92-96 (1992).

[10] **Turk, M.** and **Pentland, A.**, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, **3**(1): 71-86 (1991).

[11] **Sung, K. K.** and **Poggio, T.**, "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**(1) : 39-51, Jan. (1998).

[12] **Rowley, H., Baluja, S.** and **Kanade**, **T.,** "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**(1): 23-38, Jan. (1998).

[13] **Osuna, E., Freund, R**. and **Girosi, F.,** "Training Support Vector Machines: An Application to Face Detection" , *Proc. IEEE Conf. Computer Vision and Pattern Recog.*, PP: 130-136 (1997).

[14] **Schneiderman, H.** and **Kanade, T.**, "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, PP : 45-51 (1998).

[15] **Rajagopalan, A., Kumar, K. Karlekar, J., Manivasakan, R., Patil, M., Desai, U., Poonacha**, **P.** and **Chaudhuri, S.**, "Finding Faces in Photographs," *Proc. Sixth IEEE Int'l Conf. Computer Vision*, PP: 640- 645, (1998).

[16] **Lew, M.S.**, “Information Theoretic View-Based and Modular Face Detection”, *Proc. Int’l Conf. Automatic Face and Gesture Recognition*, PP: 198-203 (1996).

[17] **Colmenarez, A.J.** and **Huang, T.S.,** “Face Detection with Information- Based Maximum Discrimination”, *Proc. IEEE Conf. Computer Vision and Pattern Recog.*, PP: 782-787 (1997).

[18] **Pietrowcew, A.**, " Face Detection in Color Images Using Fuzzy Hough Transform", *OPTO-ELECTRONICS Review*, **11**(3), PP: 247-251 (2003).

[19] **Taylan, O.,** " Neural and Fuzzy Model Performance Evaluation of a Dynamic Production System", *Int. J. of Production Research*, **44** (6/15) :1093-1105(13), March (2006).

[20] **Lin, C.T.** and **Lee, G.,** ''*Neural Fuzzy Systems''*, Prentice Hall Int., (UK), London (1996).

خوارزم لاكتشاف الصور الأمامية للوجه باستخدام المنطق الغامض

رمضان تاسالتين، وعثمان تايلان

قسم الهندسة الكهربائية، جامعة القصيم - القصيم، و قسم الهندسة الصناعية،
كلية الهندسة، جامعة الملك عبدالعزيز - جدة – المملكة العربية السعودية

*المستخلص***.** تصف هذه الورقة طريقة جديدة لاكتشاف وجود أوجه بشرية في صورة، بغض النظر عن أحجامها وأماكنها ووضعياتها. والمعلومات الأساسية لتطوير مصنف يستخدم المنطق الغامض لاستخلاص بقعة مرشحة من الصورة تحوي منظر وجه هى لون البشرة وحدود الوجه. في البداية يتم استخلاص بقعة مرشحة لاحتوائها على وجه باستخدام معلومات لون البشرة، ثم يتم تقسيمها إلى أربع مناطق، وحساب حدود الوجه تقريبياً باستخدام معادلات متعددة الحدود. ثانيا، تم استخلاص معادلات متعددة الحدود مرجعية باختبار عدد كبير من الأوجه المختلفة ، ثم تمت المقارنة بين المعادلة متعددة الحدود المحسوبة والمعادلات المرجعية. وكنتيجة لذلك تستخدم الانحرافات بين المعادلات متعددة الحدود المحسوبة والمرجعية المستخدمة بواسطة المصنف المنطقي الغامض لتحديد ما إذا كانت البقعة، وبالتالي الصورة، تحتوي على وجه بشري أم لا.

كلمات مفتاحية: اكتشاف وجه – معلومات حدود الوجه – نظام استدلال منطقي غامض – مصنف منطقي غامض.