Frontal-view Face Detection in The Presence of Skin-Tone Regions Using a New Symmetry Approach

El_Sayed M. Saad
Faculty of Engineering, Helwan University, Egypt

Mohiy M. Hadhoud
Faculty of Engineering, Helwan University, Egypt.

Moawad I. Moawad , Mohamed El_Halawany ,and Alaa M. Abbas
Department of Electronics and Electrical Communications, Faculty of Electronic Engineering, Menouf, 32952, Menoufia University, Egypt.

ABSTRACT

In this paper, an efficient algorithm for detecting frontal-view faces in color images is proposed. The proposed algorithm has a special task; it detects faces in the presence of skin-tone regions such as human body, clothes, and background. Firstly, a pixel based color classifier is applied to segment the skin pixels from background. Next, a hybrid cluster algorithm is applied to partition the skin region. It is well known that the frontal face is symmetrical; therefore we introduce a new symmetry approach, which is the main distinguishing feature of the proposed algorithm. It measures a symmetrical value, searches for the real center of the region, and then removes the extra unsymmetrical skin pixels. The cost functions are adopted to locate the real two eyes of the candidate face region. Finally, a template matching process is preformed between an aligning frontal face model and the candidate face region as a verification step. We have tested our algorithm on 200 images from different sets. Experimental results reveal that our algorithm can perform the detection of faces successfully under wide variations of captured images.

Keywords: face detection, image segmentation, clustering, cost functions, symmetry approach.

1. INTRODUCTION AND BACKGROUND

Automatic face detection is an attractive research area. Its importance is due to its vital wide applications such as, criminal identification, security, surveillance, and intelligent human capture interaction. Human face has lots of variations of image appearance, such as lighting conditions, different size, and simple/complex background either in still images or videos. Therefore, face detection is a great challenging task that should be overcome by engineers and scientists. Over the past years, the dominant mode of images was the gray mode, therefore many researchers proposed algorithms in simple background or complex background for gray images [1-10]. Recently, the advanced technology makes easier to handle the color images by digital cameras, scanners, higher speed PCs with larger storage capacity, and broadband networks. As a result, color images have became the dominant mode and many researchers move their interest toward color images [11-13]. In general, the automatic face detection algorithms can be classified into two categories. The first category is based on the computation of geometric relationship among facial features. The second category is based on template matching. Human face detection algorithm has attracted the attention of many researchers. Jeng et al. [14] proposed an approach for detecting facial features. The first step of this approach was enhancement the contrast of the input image by using a boost filter. A matching process started by randomly selecting two facial features as the eyes, and then a weighted evaluation function was computed, if the result was larger than a certain threshold, these two features accepted as eyes. The drawbacks of this approach are the limitations of face size; not smaller than 80x80, and the image must contain only one face. Lin and Fan [15] presented a triangle-based approach for the detection of human faces. They started from the fact that the centers of two eyes with the center of mouth form an isosceles triangle for front view face image. And, the center of one eye, the center of ear hole, and the center of the mouth form a right triangle for side view face image. They extracted the potential face regions from the input image, thereafter any 3 centers of different blocks form an isosceles triangle were detected. Weighting mask function was applied to decide whether a potential face region contains a face. The algorithm failed to deal with too dark images or occluded eyes. Wu and Zhou [16] introduced a face selector method. Firstly, the eye was segmented by finding regions that are roughly as large as real eyes and darker than their neighborhoods. If two eyes placement was consistent with anthropological characteristic of human eyes, they hypothesized the regions as eyes. Some cases in which the face selector failed: one eye was near the image border, presence of glasses, dark images, and high rotation angle. Shih and Chuang [17] proposed an approach for extracting human head by high threshold image, and extracting facial features by low threshold image. An elliptical model was used to repair Low contrast of chin and to trace face boundary, then a geometric face model was used to locate facial features. Their approach valid only with simple background, failed when dealing with occluded eye, and the presence of multiple faces per image. Saber and Tekalp [18] proposed a frontal-view face detection, facial feature extraction using color, shape, and symmetry based cost functions. A classifier was used to mark each pixel as a skin/ non-skin pixel. Then, symmetry based cost functions were utilized to search the center of the eyes, tip of nose, and center of mouth within ellipses skin regions whose aspect ratio was similar to that of a face. This algorithm deals only with head and shoulder images with simple background. Cai and Goshtasby [19] proposed a method for detecting human faces in color images. The maximum likelihood was computed for each pixel to transform the color image to gray image. Then, the obtained gray image was segmented to skin/non-skin regions by using a threshold technique. A face model was used in a template matching process to detect faces within skin regions. The limitations of their method are: cannot detect small faces, and faces showing something between frontal and side views. Wei and Sethi [20] proposed face detection for image annotation. They used the presence of skin-tone pixels coupled with face specific features to locate faces in images. An iterative region partitioning procedure is used to generate candidate face regions. The iterative region partitioning procedure was started with randomly dividing a region into n clusters. If the size filter and shape filter failed to extract faces, the system repeats itself with another n, which leads to large time consumption, and the detection rate of their system was only 70% - 80%. Hsieh et al. [21] proposed a statistic approach to detecting human faces in color nature scene. The proposed algorithm was started by a skin/non-skin color classifier algorithm to obtain a skin color map, and then a splitting algorithm was used to separate the facial and non-facial areas in the color map. An elliptical model and threshold technique were used to crop the real human faces and confirm the faces, respectively. There is a condition to detect faces successfully by their algorithm; the face must not occlude by an object. Wong et al. [22] proposed a robust scheme for live detection of human faces in color images. Their algorithm was deigned to identify skin color pixels reliably under varying lighting conditions. The skin color regions were then clustered and verified as human face regions. They used eigenmask to improve the detection rate of the algorithm. This algorithm had a detection rate of 93.39% when eigenmask enabled and 87.22% when eigenmask disabled.

All previously mentioned algorithms fail when dealing with faces that are connected to another skin-tone region. Except, the system proposed by Wei and Sethi [20] as discussed previously. The proposed algorithm in this paper deals with this problem successfully and the results are promising. Also, a comparison between Wei and Sethi [20] algorithm and our proposed algorithm is performed to demonstrate the power ours.

The rest of this paper is organized as follows. Section 2, introduces the skin/non-skin color classifier. The clustering procedure is described in section 3. Section 4 proposes the symmetry approach. Section 5 presents an approach to locate eyes in skin regions. The experimental results are demonstrated in section 6. Finally the conclusion is addressed in section 7.

2. SKIN/NON-SKIN COLOR CLASSIFIER

In this section, segmentation of skin pixels from input image is introduced. The input image is RGB format, which is sensitive to lighting conditions, because the brightness and color information are coupled together. Therefore, it is not suitable for color segmentation under unknown lighting conditions. Therefore, color system transformation is needed for skin color segmentation. The color format systems are including RGB, HSV, YES, YCbCr, and etc. It is generally agreed that there is no single color format system that is suitable for all color images [21]. Here, the YCbCr color system is adopted because: it is a hardware oriented color model, and the output of digital cameras is usually either YCbCr or RGB format [22]. The color system transformation from RGB to YCbCr is define as follows:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} +0.299 & +0.587 & +0.114 \\ -0.169 & -0.332 & +0.500 \\ +0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

Where Y represents the luminance component. While Cb and Cr represent the chrominance components of a color image. The color distribution of skin colors of different people was found to be clustered in a small area of the chromatic color space, as shown in Fig.1. Although skin colors of different people appear to vary over a wide range, they differ much less in color than in brightness. In other words, skin colors of different people are very close, but they differ mainly in intensities. As a result, the Y component is discarded because it contains brightness information. But, Cb and Cr components are used because they contain the color information. A manually selected skin samples from color images were used to determine the color distribution of human skin in chromatic color space. Our samples were taken from persons of different ethnicities: Asian, Caucasian and African. As the skin samples were extracted from color images, the skin samples were filtered using a low-pass filter to reduce the effect of noise in the samples. As shown in Fig.1, the color histogram revealed that the distribution of skin color of different people are clustered in the chromatic color space and a skin color distribution can be represented by a Gaussian model G(m,std), where: m is the mean, and std is the standard deviation. The detection window for skin color was determined based on the mean and standard deviation of Cb and Cr component. It can define as follows:

$$t_{Cb\,Lower} = m_{Cb} - std_{Cb} \times p \tag{2}$$

$$t_{Cb\,Upper} = m_{Cb} + std_{Cb} \times p \tag{3}$$

$$t_{Cr\,Lower} = m_{Cr} - std_{Cr} \times p \tag{4}$$

$$t_{Cr\,Upper} = m_{Cr} + std_{Cr} \times p \tag{5}$$

p : factor determines the width of the gaussian envelop.

$$Cb_{Skin} = \begin{cases} 1 & t_{Cb\,Lower} < Cb < t_{Cb\,Upper} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$Cr_{Skin} = \begin{cases} 1 & t_{Cr\,Lower} < Cr < t_{Cr\,Upper} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

For better segmentation, the intersection between two components only is considered as follows:

$$f = \begin{cases} 1 & \text{if } [Cb_{Skin} \cap Cr_{Skin}] \text{ is True} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

f: is the binary skin color map output image.

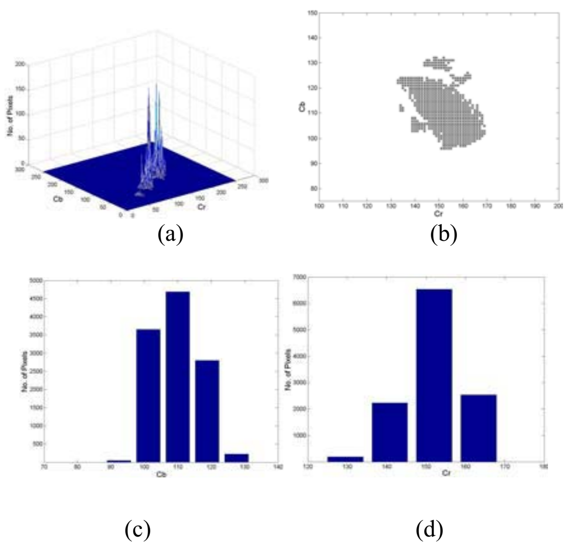


Fig.1 a) 2 D histogram of Cb Cr, b) Cb-Cr plan, c) 1D histogram of Cb, and d) 1D histogram of Cr

In fact, the obtained binary image doesn't contain only the human skin color, but also contains objects, which have the same color range as that of the human skin color. Therefore, a morphological operation as area open [23,24] is then applied to remove the small blobs that may be appeared in the binary image. From careful observation, we have found that the human skin color pixels region is the largest area of pixels in the binary image. So, a labeling process is applied to the binary image and the corresponding area of each patch is computed and sorted according to its areas. Then, the first area (largest area) only is selected under one condition; the percentage of the first area size to the second area size is greater than 1.3 (experimentally). If we face an image contains multiple separate faces (football team image) the largest area selection step is automatically discarded because the patches are almost had the same area size. Hence, the algorithm will jump to the next step which is the clustering algorithm. Fig.2 shows an example for the above procedure; a-the original image contains a face that is connected to another skin-tone region (an arm, clothes, and body), b-the binary skin color image, and c-images after applying morphological operation and the largest area selection.

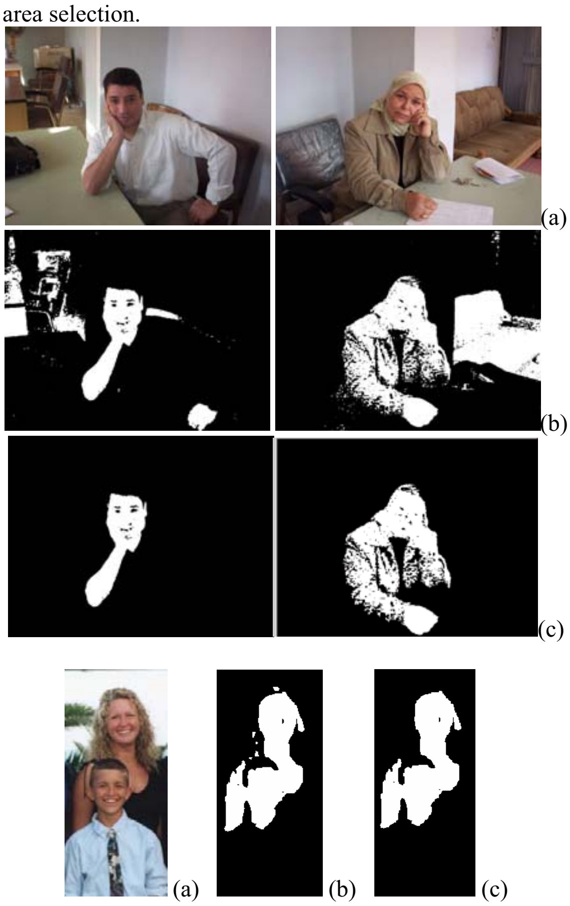


Fig. 2. (a) Original images, (b) the binary skin color map, and (c) after area open and largest area selection.

3. CLUSTERING ALGORITHM

As mentioned in previously, any face detection algorithm will fail to detect the face from the obtained binary image because the face is connected to another skin-tone region. Hence, a special treatment is considered to extract the face from the obtained binary skin color image. To overcome this problem, the skin region is clustered into sub-regions. Several clustering algorithms reported in the literature. K-means clustering algorithm [24-26] is adopted to cluster the binary image, but the K-means needs to predetermine the number of clusters. Therefore, maximin clustering algorithm [24,25] is adopted to automate the determination of the number of clusters. As a result, we have used clustering algorithm combines both of them. It determines the number of clusters and its corresponding centers automatically. Thereafter, this number and its corresponding centers are fed to K-means cluster algorithm. The advantages of this approach over the traditionally K-means are:

1-Automatic determination of number of clusters, and initially corresponding centers,

2-Reduction of the number of iteration required for convergence.

The algorithm is carried out as follows:

Let the given pixels: $f = \{f_i\}_{i=1}^{n} \quad 1 \le i \le n$,

where n is the number of pixels, and

$\{c_j\}_{j=1}^{k} \quad 1 \le j \le k$, the cluster centers, where k is the number of cluster centers found.

Step 3.1: Arbitrarily, let $c_1 = f_1$

Step 3.2: Determine the farthest pixel from f_1, and call it cluster center c_2

$$\|c_2 - c_1\| = \max_{2 \le i \le n} \|f_i - c_1\|$$

Step 3.3: Compute the distance from each remaining pixels to c_i, for every one of these computations, select maximum of the minimum distance

$$f' = f - \{c_j\}, \text{and}$$

$$d = \|f_{io} - c_{jo}\| = \max_{f_i \in f'} \min_{1 \le j \le k} \|f_i - c_j\|,$$

where $1 \le j_o \le k$, and $f_{io} \in f'$

Step 3.4: If the distance d is an appreciable fraction of distance between cluster center $\{c_j\}$, we call corresponding pixel a new cluster center.

$m = \overline{\|c_i - c_j\|}$ (arithmetic mean), where $1 \le i,j \le k$, $i \ne j$

$$\text{if} \begin{cases} d \ge t_c.m \text{ (newcluster) go to step 3.3} \\ \text{otherwise (no more cluster) go to step 3.5} \end{cases}$$

where, t_c is a threshold value which determines whether a new cluster should be created.

Step 3.5: Assign each remaining pixels to its nearest cluster center.

for each $f_i \in f'$: $\|f_i - c_j\| = \min_{1 \le j \le k} \|f_i - c_j\|$

Step 3.6: Take the pixel mean for each cluster. Those means can then be used as the new cluster centers.

For $1 \le j \le k$, replace c_j by: $c_j = \frac{1}{z_j}\left(\sum_{j=1}^{z} f_j\right)$,

where, z is the number of ixels of cluster j.

Step 3.7: Redistribute the pixels f' among the c_j(iter) cluster domain, using the relation:

$$f \in s_j(\text{iter}) \text{ if } \|f - c_j(\text{iter})\| \le \|f - c_i(\text{iter})\|$$

where, s_j denotes the set of pixels choose cluster center is c_j, and $1 \le i,j \le k$, $i \ne j$

iter : is the number of iterations needed for convergence.

Step 3.8: Compute the new cluster centers

$$c_j(\text{iter}+1) = \frac{1}{z_j}\left(\sum_{j=1}^{z} f_j\right)$$

such that the sum of the squared distance from all pixels in s_j to the new cluster centers is minimized. In other words, the new cluster center c_j is computed so that performance index is minimized.

$$I_j = \sum_{f_i \in s_j(iter)} \left\| f_i - c_j(iter+1) \right\|^2, \quad j = 1,2,\ldots,k$$

Step 3.9: Check for convergence that will be occurred if none of cluster centers are changed:

$$\text{if}\begin{cases} c_j(iter+1) = c_j(iter)\; 1 \le j \le k, \text{stop and terminate} \\ \text{otherwise go to step 3.7. End} \end{cases}$$

Fig. 3 shows results of the described clustering algorithm. Here, the algorithm automatically decides to break the skin region (R) into sub-regions (R_j , j = 1,2,…,k).

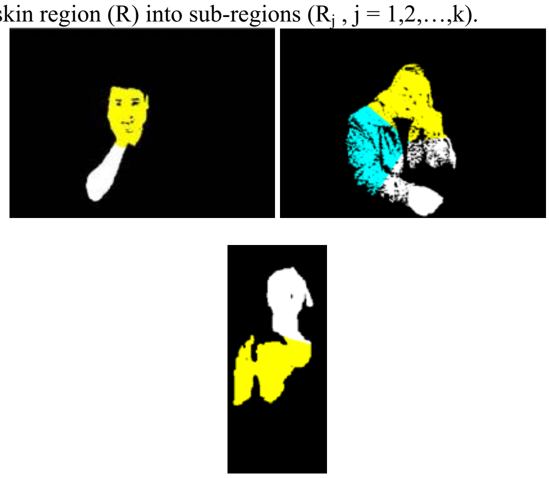


Fig.3. Clustering process

4. NEW SYMMETRY APPROACH

At this stage, we have started from the fact that the vertical human face is symmetrical. The symmetrical shape of a region R_j that has been produced in the previous section is measured, and then the unsymmetrical pixels from a region R_j under test are removed. Here, we have used a searching technique to find the real center of the region R_j, which gives the maximum symmetrical value. The symmetrical value is measured as follows:

$$S = \frac{\text{Number of symmetrical pixels}}{\text{Total number of pixels}} \tag{9}$$

A pixel is said to be vertical symmetric, if it exists at (x,y), and (-x,y). Fig. 4 shows an example; the goal is to calculate the symmetrical values at vertical lines which named line 1, line 2, and line 3.

$$S_{\text{at vertical line 1}} = \frac{3}{10}, S_{\text{at vertical line 2}} = \frac{8}{10}, \text{ and } S_{\text{at vertical line 3}} = \frac{1}{10}$$

The symmetrical value (0.8) at vertical line 2 is accepted, and then the unsymmetrical pixels are removed as shown in Fig. 4 (right). Note that, S is varying from 0 to 1.

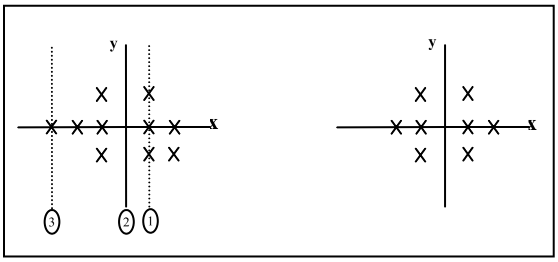


Fig. 4.
Left: the symmetrical value is measured at lines 1,2, and 3.
Right: the final result corresponding to the max value.

The details of symmetrical measurement algorithm are stated below:

Step 4.1: Fill the holes that fall inside a region R_j
Step 4.2: Calculate the center of gravity (c_x , c_y) of the region R_j.
Step 4.3: Find a real center, which gives the maximum vertically symmetric value as follows: Refer to Fig. 5

Step 4.3.1: Measure the symmetry at pixels which marked by "1" (three pixels apart from c_y). Among these measurements, we choose the largest value. Suppose at c_{y1}.
Step 4.3.2: Measure the symmetry at pixels which marked by "2" (two pixels apart from c_{y1}). Among these measurements, we choose the largest value. Suppose at c_{y2}.
Step 4.3.3: Measure the symmetry at pixels which marked by "3" (one pixels apart from c_{y2}). Among these measurements, we choose the largest value. Suppose at c_{y3}.

Step 4.4: If the final symmetrical value is larger than a certain threshold t_S : accept the region R_j, otherwise neglect it.
Step 4.5: The unsymmetrical pixels are removed from the accepted region R_j.
Step 4.6: Use morphological operation area open to remove small blob. End.

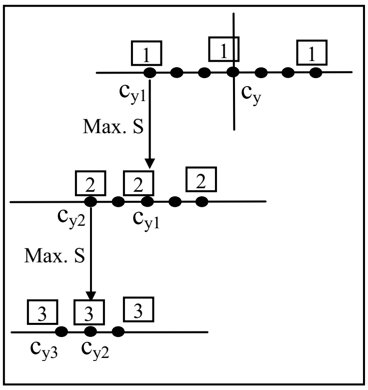
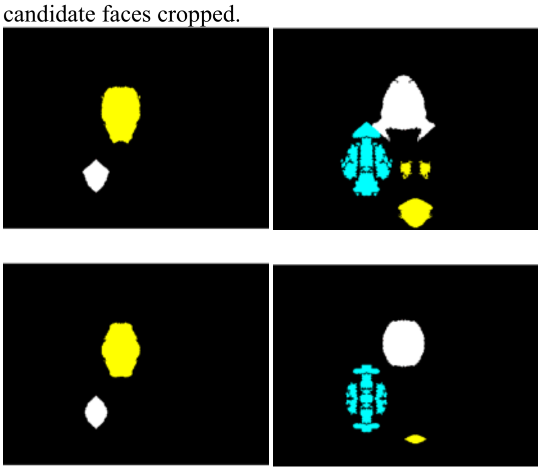


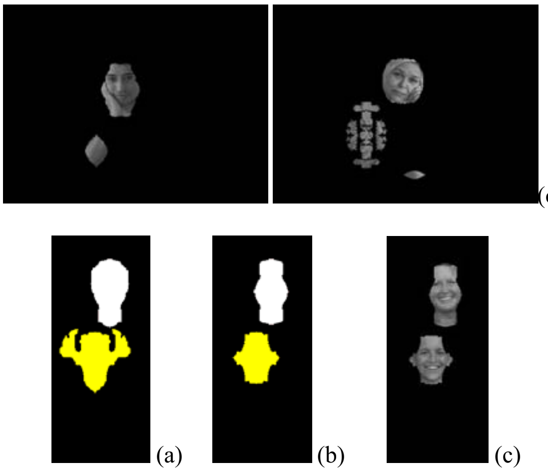
Fig. 5. The Searching technique for the real center

From the geometry of the human face, it has an elliptical shape; the above approach is repeated for the horizontal axis in sequence after the vertical axis. Fig. 6 illustrates the results of the above approach: a-removing unsymmetrical pixels around vertical axis, b-removing unsymmetrical pixels around horizontal axis, and c-the candidate faces cropped.



(a)

(b)



(c)

(a) (b) (c)

Fig. 6. (a)Vertical symmetry, (b) horizontal symmetry, and (c) the candidate faces cropped.

5. LOCATING THE TWO EYES

In this section, the two eyes will be located. Then, a frontal-view face model and the candidate face region are used in a template matching process as a verification step. Eyes are the most characteristics regions on the face, therefore, existence of the two eyes are evidences that the candidate region indeed a face. Area of the eye is usually darker than the rest of the candidate face region. This observation leads us to the following approach to detect the presence of eyes:

First, A median filter is applied to smooth the candidate face region (in gray mode):

f_s=med filter (f,3x3) (10)

Furthermore, an average filter is applied with a threshold to extract the possible eyes holes that are the lowest intensity values of its neighbors as follows:

$$f_{av}(m,n) = \frac{\sum_{i=\frac{1-h}{2}}^{\frac{h-1}{2}} \sum_{j=\frac{1-w}{2}}^{\frac{w-1}{2}} k(i,j).f_s(m+i,n+j)}{((w.h)-1)} \tag{11}$$

where w,h are the width and height (odd numbers) of the filter window K used in Eq. (11).

A thresholding technique is applied to segment the candidate face region into eye/non-eye pixels:

$$F = \begin{cases} 0 & \text{if } f_s \le (t_e . f_{av}) \\ 1 & \text{otherwise} \end{cases} \tag{12}$$

where the threshold t_e depending on lighting conditions. If the region contains no holes (calculated by the Euler number) the algorithm will discard the region. Fig. 7 shows the results of applying the above approach.

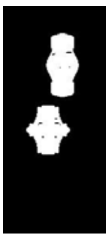


Fig.7. The possible eyes extraction

The cost functions [18] are adopted to locate the real eyes from possible eyes that have been extracted at last step. The centroid of each eye is employed to examine its location as follows:

Let c_{xF} , c_{yF} indicate the centroid of the candidate face region, and c_{xe} , c_{ye} the centroid of candidate eye hole. Refer to Fig.8.

$$\text{If} \begin{cases} c_{xe} < c_{xF} & \text{accept this eye} \\ \text{otherwise} & \text{neglect this eye} \end{cases} \tag{13}$$

$$\text{If} \begin{cases} c_{ye} < c_{yF} & \text{accept this eye as left eye } c_{yeL} \\ c_{ye} > c_{yF} & \text{accept this eye as right eye } c_{yeR} \end{cases} \tag{14}$$

The cost functions are applied to locate one left eye with one right eye [18]:

$$CF_{LR}^{1} = abs(c_{xLe} - c_{xRe}) \tag{15}$$

$$CF_{LR}^{2} = abs(abs(c_{yLe} - c_{yF}) - abs(c_{yRe} - c_{yF})) \tag{16}$$

$$CF_{LR}^{3} = abs(abs(c_{xLe} - c_{xF}) - abs(c_{xRe} - c_{xF})) \tag{17}$$

$$CF_{LR}^{4} = abs(c_{xLe} - c_{xF}) + abs(c_{xRe} - c_{xF}) \tag{18}$$

$$CF_{LR}^{5} = (c_{xLe} - c_{xF}) + (c_{xRe} - c_{xF}) \tag{19}$$

The weighted combination of these cost functions is:

$$CF_{LR} = \sum_{i=1}^{5} w_i . c_{LR}^{i} \tag{20}$$

Its minimum represents the two holes within the region that are most likely the eyes.

The locations of the eyes are shown in Fig. 9.

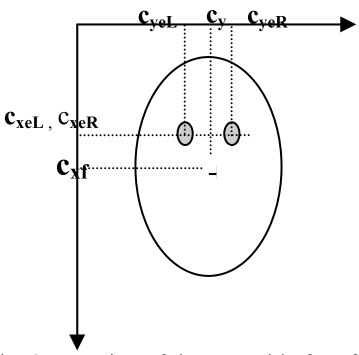


Fig. 8. Location of the centroid of Left eye and right eye respect to face centroid likely eyes inside regions.

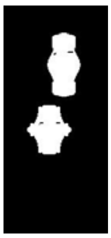


Fig. 9. Location of the most

The rotation angle of the accepted region and the distance between the centers of two eyes are computed by the following relations:

$$\theta = \tan^{-1}\left(\frac{c_{xLe} - c_{xRe}}{c_{yLe} - c_{yRe}}\right) \tag{21}$$

$$d = c_{yRe} - c_{yLe} \tag{22}$$

Note that, the distance d is measured after fixing the region to be vertical.

Once the rotation angle and the distance are computed, a matching process is performed by:

1-Scaling a model so that it has the same width as that of the region (distance between centers of the eyes in both of them is the same).

2-Orienting a model so that it has the same orientation as that of the region. The model is readjusted so that its eyes fall on the region's eyes.
Then cross-correlation function between the candidate face region and aligning face model is computed:

$$C = \frac{\sum_i \sum_j (f_f - \overline{f_f}).(f_m - \overline{f_m})}{\sqrt{\left[\sum_i \sum_j (f_f - \overline{f_f})^2\right]\left[\sum_i \sum_j (f_m - \overline{f_m})^2\right]}} \quad (23)$$

where f_f is the candidate face region image, f_m is the model face image, and $\overline{f_f}$ and $\overline{f_m}$ are the corresponding mean of the two images.
A large correlation value than a threshold t_m means that the algorithm decides a human face is detected, or the region does not a human face.

6. EXPERIMENTAL RESULTS

The performance of the proposed algorithm is tested on three face data sets. In the first set, images are captured by Kodak Easy Share CX6200 digital camera. Each image contains at least one human face with image size 616x816. Simple/complex background, varying lighting conditions, and various faces size are taken into account when the images have been captured. In the second set, images are downloaded from the Internet with wide range of variations. In the third set, images (frames) are cropped from movie files (mpeg files). All experiments are performed on Pentium II (350 MHz), and all the codes are written in Matlab 6. In section 3, the experimental results reveal that the best value of t_c is set to be 0.7.
Table 1 shows a comparison between the proposed hybrid clustering algorithm and the traditional K-means clustering algorithm. In section 4, t_S is set to be at least 0.5 to accept the region R_j. The results show the best value of t_e is between 0.7 and 0.8 depending on the lighting conditions. In section 5, t_m that decides whether the candidate face region is a face or not is set to be 0.85.
As shown in table 2, the proposed algorithm is tested on 200 images (from three sets), the total number of faces in all images were 240 faces. The faces that have been detected correctly using our algorithm were 229 faces (detection rate = 95.5 %). The faces that were missed by the proposed algorithm were due to the following reasons:

1- They have been taken under poor lighting conditions.
2- An object occludes one or the two eyes.
3- Faces have high rotation angle.

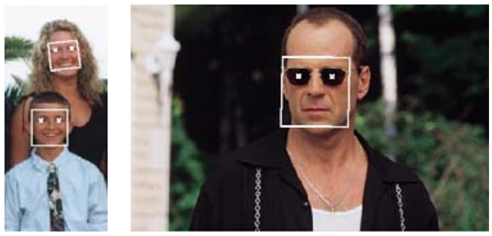
These reasons lead to certain conditions should be verified to detect faces successfully by our algorithm:

1-The faces should be vertical or within relatively small rotation angle for good performance of symmetry approach.
2-The eyes should not been occluded by an object.

Table 3 shows a comparison between the proposed algorithm and Wei and Sethi [20] algorithm (related algorithm). We have chosen their algorithm because it is very close to the proposed algorithm. It should be noted that the Matlab codes are usually 9 or 10 times slower than c/c++ equivalents [27]. Fig. 10 shows some results of the correctly detected faces and corresponding eyes locations. It is obvious that our algorithm detects accurately and efficiently faces and locates eyes in the presence of skin-tone regions.

(a) faces are connected to an arm and clothes

(b) faces are connected to arms



(c) faces are connected to human body

face is connected to another face : left large size, right small size

Fig. 10. Sample of results of the proposed algorithm
(a) images captured by digital camera, (b) images cropped from movie file, and (c) images downloaded from Internet

Table 1. A comparison between traditional K-means and hybrid clustering algorithm.

No. of Cluster	Traditional K-means		Hybrid algorithm	
	No. of iter	Process time	No. of Iter	Process time
2	8	26 s	4	11 s
3	11	32 s	6	17 s
4	15	37 s	9	23 s
5	23	49 s	13	28 s

Table 2. Detection performance of the proposed algorithm.

Set type	No. of imag	No. of fac.	Corr. Detec faces	Mis. Fac.	Fals det.. fac.	Ave Pros. time in sec./ face	Det. Rate %
Dig. cam	100	116	114	2	0	103	98.2
Inte net	60	76	71	5	2	86	93.4
Fra m	40	48	44	4	1	99	91.6
Tot	200	240	229	11	3	96	95.5

Table 3. A comparison between the proposed algorithm and Sethi algorithm.

Alg.	No. of imag	No. of fac.	Corr. Detec faces	Mis. Fac.	Fals det.. fac.	Ave Pros. time in sec./ face	Det rate %
Prop.	200	240	229	11	3	96	95
Sethi	200	240	197	43	5	119	82

7. CONCLUSION

The traditional face detection algorithms fail to detect faces that are connected to another skin-tone region, which leads to low detection rate. The presence of skin-tone regions makes the detection problem more difficult and needs a special treatment to detect faces in such case. The contribution of this paper is detection of faces in the presence of skin-tone regions. The proposed algorithm starts from classifying each pixel into skin/non-skin pixel. A hybrid cluster algorithm is applied to break a skin region into smaller skin patches. A new symmetry approach is used to measure the symmetrical value and find the real center of the face. Then removes the unsymmetrical neighbors skin pixels. The existence of the two eyes is an evidence to decide whether the candidate face region is a face or not. So, we have adopted cost functions to locate the real two eyes from the possible eyes that have been obtained by the average filter. The proposed algorithm considers only frontal-view faces images with low rotation angle. When the faces are highly rotated the symmetry approach cannot extract the real face region. The experimental results reveal that the proposed algorithm is efficient to detect faces that are connected to another skin-tone region (detection rate = 95.5 %) from different three data sets. Our future work is developing a face recognition system, whose first step is the face detection. So, the proposed algorithm will be used as a first step when the faces are connected to another skin-tone region.

8. REFERENCES

[1] H.A. Rowley, S. Bluja, T. Kanade, ”Neural network-based face detection”, IEEE Transactions, Pattern Analysis and Machine Intelligence 20 (1)(1998) 23-38.
[2] K.-K. Sung, T. Poggio, “Example-based learning for view-based human face detection”, IEEE Transactions, Pattern Analysis and Machine Intelligence 20 (1)(1998) 39-51.
[3] I. Craw, H, Ellis, J.R, Lishman, “Automatic extraction of face-features’’, Pattern Recognition Letter 5 (1987) 18-187.
[4] D.W. Purnell, C. Nieuwoudt, E.C. Botha, “Automatic face recognition in a heterogeneous population’’, Pattern Recognition Letter (1998) ()1067-1075.
[5] C.L. Huang, C.W, Vhen, “Human facial feature extraction for face interpretation and recognition’’, Pattern Recognition 25 (12) (1992) 1435-1444.
[6] C.H. Lee, J.S. Kim, K.H Park, “Automatic human face location in a complex background using motion and color information”, Pattern Recognition 29 (11) (1996) 1877-1889.
[7] G. Craw, X, Li, “Toward a system for automatic facial feature detection”, Pattern Recognition .26 (12) (1993) 1739-1755.
[8] K.C. Yow, R. Cipolla,” Feature-based human face detection,” Image Vision Computer 15 (19997) 713-735.
[9] G. Yang, T.S. Huang, “Human face detection in a complex background”, Pattern Recognition 27 (1) (1994) 53-63.
[10] L. Xu, M. Jackowski, A. Goshtasby, D. Roseman, S. Bines, C. Yu, A. Dhawan, A. Huntly, “Segmentation of Skin Cancer Images”, Image and Vision Computing 17 (1) (1999).
[11] Q. Chen, H. Wu, M. Yachida, “Face detection by fuzzy pattern matching”, proceedings of the Fifth International Conference on Computer Vision, 1995, pp. 591-596.
[12] Y.Miyake, H. Saitoh, H. Yaguchi, N. Tsukada, “ Facial pattern detection and color correction from television picture and newspaper printing”, Journal of Image Technology 16 (5) (1990) 165-169.
[13] K. Sobottka, I. Pitas, “Segmentation and tracking of faces in color images”, Proceeding of the second International Conference on Automatic Faces and Gesture Recognition 1996, pp. 236-241.
[14] Shi-Hong Jeng, Hong Yaen Mark Liao, Chin Chuan han, Ming Yang Chern, and Yao Tsorng Liu, “ Facial feature detection using geometrical face model: an efficient approach”, Pattern recognition, Vol. 31, No. 3, pp. 273-282, 1998.
[15] Chiunhsiun Lin, Kuo-Chin Fan, “Triangle-based approach to the detection of human face”, Pattern Recognition 34 (2001) 1271-1284.
[16] Jainxin Wu, Zhi-Hua Zhou, “Efficient face candidates selector for face detection”, Pattern Recognition 36 (2003) 1175-1186.
[17] Frank Y. Shih , Chao-Fa Chuang, ”Automatic extraction of head and face boundaries and facial features”, Information Sciences 158 (2004) 117-130.
[18] Eli Saber, A. Murat Tekalp, “ Frontal-view face detection and facial feature extraction using color, shape and symmetry cost functions”, Pattern Recognition Lett. 19 (1998) 669-680.
[19] J. Cai, A. Goshtasby, “Detecting human faces in color images”, Image and Vision Comp.18 (1999) 63-75.
[20] Gang Wei, Ishwar K. Sethi, “Face detection for image annotation”, Pattern Recognition lett. 20 (1999) 1313-1321.
[21] Ing-Sheen Hsieh, Kuo-Chin Fan, Chiunhsiun Lin, ”A statistic approach to the detection of human faces in color nature scene”, Pattern Recognition 35 (2002) 1583-1596.
[22] Kwok-Wai Wong, Kin-Man Lam, Wan-Chi Siu, “ A robust scheme for live detection of human faces in color images”, Signal Processing: Image Communication 18 (2003) 103-114.
[23] William K. Pratt, ”Digital Image Processing”, A wiley-interrscience publication, 1991,Second edition.
[24] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974.
[25] Menahem Friedman, Abraham Kandel, Introduction to Pattern Recognition, Imperial Cellege Press 1999.
[26] Mu-Chun Su, Chien-Hsing Chou, “A Modified Version of K-Means Algorithm with a Distance Based on Cluster Symmetry”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 6, June 2001.
[27] Kyug-Min Cho, Jeong-Hun Jang, Ki-Sang Hong,”Adaptive Skin-color Filter”, Pattern Recognition 34 (2001)1067-1073.

Received: Feb. 2006. Accepted: Jul. 2006.