Robust Profile Face Detection

Shady S. Al-Atrash and Ibrahim S. Abuhaiba

Islamic University-Gaza
Eng_Sh_Atrash@hotmail.com, Ihaiba@iugaza.edu.ps

Abstract

The objective of this paper is to propose a new algorithm in face detection that has the capabilities of detecting the face with different poses and under different conditions. This objective is obtained in different stages and using different proposed algorithms. Firstly, a robust segmentation algorithm is proposed to extract and segment the skin region from the image. Secondly, different filtering steps are applied to this segmented image to obtain the face candidate region only. After that, Feature-Based approach is used to detect the features from this candidate face which can work in real-time with minimal training in contrast to other approaches such as image-based approach. Finally, some rules is applied in order to judge if this candidate is profile face or not either the profile face is right or left.

Experimental results show that the proposed method is robust under a wide range of lighting conditions, different poses and different races. These results are taken from three different face databases. The proposed method is implemented using Matlab version 7.6 software and gives a correct detection rate reach 90 %.

Keywords: *Face Detection, Skin Segmentation, Profile Face, Complex Background, Different Lightening Conditions, Face Pose*

1. Introduction

The face plays a main role in carrying identity of persons. One of things that really admire the viewer is the human ability to recognize faces. Humans can recognize thousands of faces and identify familiar faces despite large changes in the visual stimulus due to viewing conditions, expression, aging, sex, and distractions such as glasses or changes in hair style [1]. Many recent events, such as terrorist attacks, exposed serious weakness in most sophisticated security systems. Various government agencies are now more motivated to improve security data systems based on body or behavioral characteristics, often called biometrics. Face Detection (FD) is one of the main biometric features that many researches concentrate on developing algorithms to apply it in different systems. FD is one of the fundamental techniques that enable a Human Computer Interaction (HCI) [2]. So, FD is the main step stone to all facial analysis algorithms, including face alignment, face verification/authentication, gender/age recognition and Face Recognition (FR).

The goal of FD is to determine whether or not there are any faces in the image and, if present, locate the image face. Early efforts in FD have presented as early as the beginning of the 1970s, where simple heuristic techniques [3] were used. These techniques are largely rigid since they assume ideal conditions such as plain background and frontal face. Some of the factors that make FD such a difficult task are face orientation, face size, different facial expression and feature, occlusion and lighting condition.

FD techniques require a priori information of the face, so, they can be effectively organized into two broad categories which can be used to determine the face knowledge from one person to other. The first approach is feature-based [1, 4, 5] which make explicit use of face knowledge and follow the classical detection methodology in which low level features are obtained prior to knowledge-based analysis. Typically, in this approach FD tasks are accomplished by manipulating distance, angles, and area measurements of the visual features derived from the image, while the second approach is image-based[1, 4, 6] approach which is unlike the feature-based approach, it incorporate face knowledge implicitly into the system through mapping and training schemes.

Many researches have been developed for developing good face detection algorithms. Kanade [7] presented an automatic feature extraction method based on ratios of distances (between feature points such as the location of the eyes, nose, etc.) and reported a recognition rate of between 45-75% with a database of 20 people. Brunelli and Poggio [6] compute a set of geometrical features such as nose width and length, mouth position, and chin shape. They report a 90% recognition rate on a database of 47 people. Shih et.al [8], presents a novel approach for the extraction of human head, face and facial features. They try to locate a face in a normal lighting condition and using plain background and. However, it is difficult to detect a face in a complex background and under various lighting conditions. Hsu, et. al. [9] present a new approach for FD in color image in the presence of varying lightening conditions as well as complex background which gives a good results.

We propose a new algorithm to be used in this field and give good results. Figure 1, shows the block diagram of the proposed FD process.

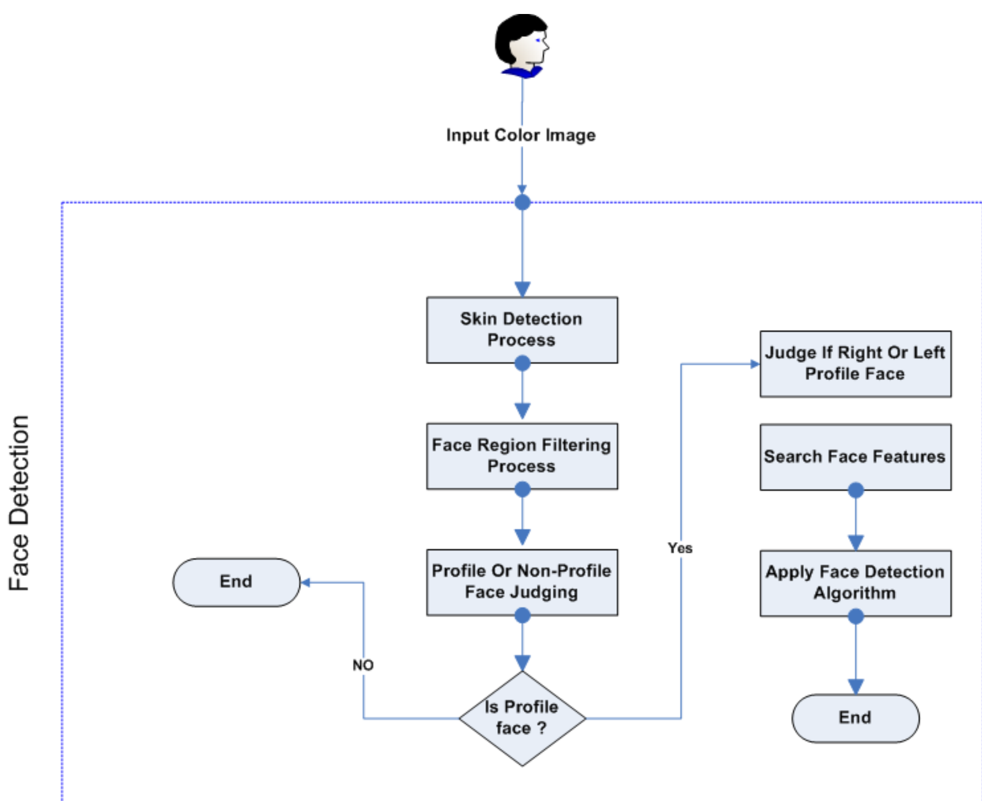


Figure 1. Block Diagram of the Proposed FD Scenario.

2. Learning Skin Color Model Parameters

In this section, we avoid the traditional approach that uses skin color thresholds for face segmentation, because this approach doesn't work efficiently in different lighting conditions. We propose a new model for skin color classifying that classify skin pixels based on statistical measures that model face variation within a wide user spectrum.

This proposed model based on using Gaussian distribution to represent a skin color cluster. The Gaussian distribution is characterized by its mean (μ) and covariance matrix (∑). Pixel color from an input image can be compared with the skin color model by computing the Mahalanobis distance. This distance measure gives an idea of how close the pixel color resembles the skin color of the model. The advantage of this statistical color model is that color variation of new users can be adapted into the general model by a learning approach.

This proposed model uses a number of skin samples taken from persons of different races (Asian, African, European) to determine the color distribution of human skin in color space. These skin samples are taken from the GTAV [10] and FEI [11] frontal face databases. We segment a total of **9,000 (40*30)** skin samples from these face databases images to be used as training examples to determine the color distribution of the skin color model. In this work, we take only red and blue chrominance since the human faces dominant colors chrominance are red and blue (cr, cb).

3. Proposed Skin Segmentation

After obtaining the previous skin color model, we can now obtain the gray likelihood of skin for every pixel (The gray value at each pixel shows the likelihood of the pixel belonging to the skin) of an image using multivariate Gaussian equation obtained using the previous color model.

If a pixel, having transform from RGB color space to chromatic color space, has a chromatic pair value of (r,b), the likelihood of skin for this pixel can then be computed as follows:

$$\textbf{Likelihood} = \textbf{exp} \{(-0.5)(x-m)^T C^{-1}(x-m)\}, \ldots\ldots\ldots\ldots.. (1)$$

where,

$x:(r,b)^T$, m : Mean and C^{-1}: Covariance Inverse Matrix

Equation 4.3 results a gray scale image between [0-255] and each pixel represents the likelihood of skin in this pixel.

3.1 Image Filtering

After segmenting the image to skin and non-skin regions, it may contain some regions that don’t belong to a face but have the same face skin color (such as backgrounds, clothes…, etc), in this step we do some candidate filtering operations to remove these irregular regions. Some of these filtering operations are already exists in image processing concepts and others are presented by us to enhance the segmentation operation. The following topics present these filtering operations.

3.1.1 Width and Height of A Region: Some segmented face images have regions with irregular shapes and they must be deleted in order to have only face region part. To remove these irregular regions we propose a filtering step that depend on determining the width and height of the region bounding rectangle and delete the regions that shape like a vertical thin bar or horizontal thin bar according to certain threshold as follows:

- If the region bounding rectangle width or height < **Rectangle_Threshold** ➔ Delete the region.

3.1.2 Solidity Property: This is not a new property, we use it to describe the solidity of the regions in the segmented image so, we can take its advantages in our segmentation algorithm. The solidity of any region in the binary segmented image is an important geometrical feature. Based on this geometrical property, we take the solidity of the face regions into consideration as follows:

- If the region solidity < = **Solidity_Threshold** ➔ Delete the region.

3.1.3 Shoulders Removal: In this step we present a technique for removing shoulders from the face image if they exist. The shoulders removal process consists of two stages: shoulders searching and shoulders elimination.

<u>Step 1</u>: Shoulders Searching

We depend on the fact that "*The shoulders part of people is wider than the head and neck parts*". we follow certain steps that lead us to judge if the image contains shoulder parts or not. First of all, a *horizontal projection* for the face skin pixels is taken as shown in Figure 2(b). Second, we determine the *middle line* that partitions the horizontal projected image to upper and lower regions, as shown in Figure 2(c). The process of determining the middle line will be shown in shoulder elimination topic. Third, depending on this line, a calculation to the number of pixels that exist below and above the middle line in done. In most cases, "*If the number of the skin pixels below the middle line part of the whole skin region larger than the number of skin pixels above the middle line then the shoulders should be exist*". Fourth, we find the ratio between upper part and lower part pixels to judge if the shoulders exists or not as follows:

- If lower/Upper > **Shoulder_Threshol** ➔ Shoulders exists.

(a) (b) (c)

Figure 2. The Horizontal Projection of a Profile Face: (a) Right Profile Image; (b) Horizontal Projection Histogram; (c) The Middle Line Partitions the Horizontal Projected Image in (b).

<u>Step 2:</u> Shoulders Elimination

The elimination step requires a reference point to start the cutting (elimination) from it. In our work, we select the *neck point* to be this reference point and the following steps used to determine the neck point pixel coordinates. Figure 3, summarize the shoulders elimination steps using an illustration example. The elimination strategy consists of four steps. First, draw a line passing between the beginning point and maximum point of the previous calculated horizontal projection values calculated in shoulders searching step. Figure 3(b) shows the image after applying all the filtering steps and Figure 3(c) shows this line. Second, project the skin pixels segmented after previous filtering steps and which are below this line on this line. Third, the pixel with the maximum distance from this line will be the neck point, and this step gives a good results in most cases and under different databases. Fourth, delete the face pixels below this neck point row. Figure 3(d) shows the shoulders elimination result.

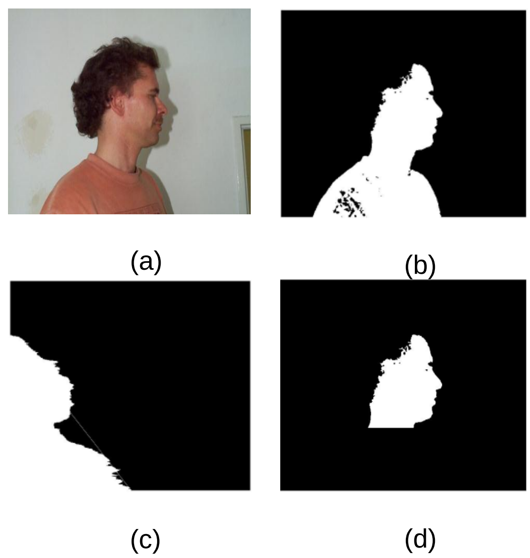


Figure 3. Successful Shoulder Removal Process: (a) Original Image; (b) Segmented Image; (c) Line Passing between the Beginning Point and Maximum Point; (d) Face after Shoulder Removal

4. Face Feature Searching

The main next step is how to detect face features that can lead us to detect the face region exactly from the still image. In our work, we focus in using the "Feature-based" approach for proposing a robust FD technique. Our proposed work depends on detecting high-level features from the image, so we are not going into detecting low level features, instead we are looking to detect high level features and these detected features will lead to a stable judge if the tested image has a face or not regardless of availability of many features that take a huge work in filtering and isolating the important features and rejecting the unnecessary features. The face has a lot of features to be detected; we work on a profile images and we take one case for discussion which is the *right profile* images. While, the *left profile* images have the same steps and we don't discuss it. The following sections show what these features are and how to detect these features in order to detect the face based on them.

4.1 Nose Point

This point is the one with the furthest right distance of the face skin region segmented previously. This point calculated according to the following steps. Step1, compute the distance from each skin region pixel (x1,y1) and the y-axis (x2,y2) according to the following distance equation:

$$\Delta x = x2\text{-}x1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$\Delta y = y2\text{-}y \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$\textbf{Distance} = \sqrt{\Delta x^2 + \Delta y^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Step2, in most cases, the point with the maximum distance will be the Nose point. Figure 4(b) shows the vertical line passing through the Nose point in the segmented image.

(a) (b)

Figure 4. The Vertical Line Passing through the Nose Point of the Face: (a) Original Image; (b) Vertical Line Passing through Nose Point

4.2 Chin Point

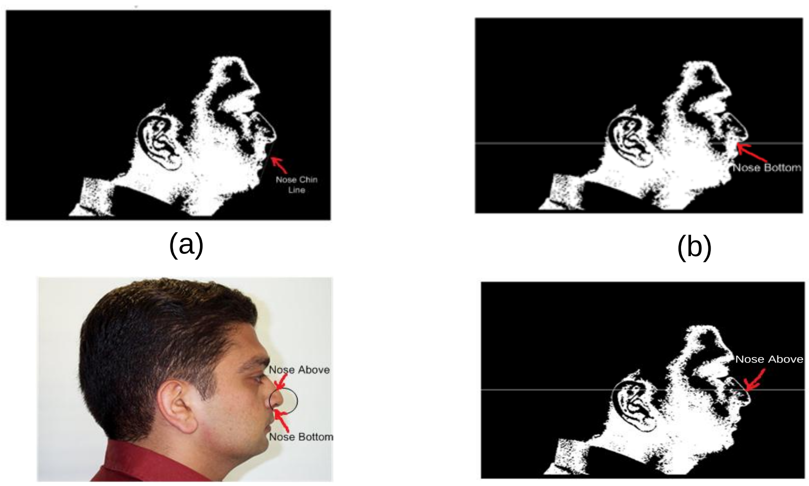
This point is calculated based on the Nose point calculated previously. The following steps show the process for detecting this point. Step 1, a rotation to the detected vertical line clockwise around the Nose point is done until the length of the intersection segment between the rotated vertical line and the segmented skin region is larger than some *chin threshold*. Step2, calculate the distance from each pixel (x1,y1) of the segmented skin region below the Nose point and to the right side of the rotated vertical line (x2,y2) resulted in step1, according to the Euclidean distance equations 1-3. Step3, in most cases, the point with the maximum distance is considered as the Chin point. Figure 5(c) shows the line passing through the chin point.

(a) (b) (c)

Figure 5. Determining the Nose Point: (a) Profile Image; (b) Rotated Vertical Line by 25o around the “Nose Point” which satisfies the “Chin Threshold”; (c) The Detected Chin Line Row.

4.3 Nose Bottom Point

This point is the one that resides below the nose point and the nostrils reside above it. The following procedure is used to detect this point. Step1, determine the line (*NCLine*) passing between Nose point and Chin point. In Figure 6 (a), the line passes between nose point and chin point is marked by red arrows. Step 2, Calculate the Euclidean distance from each pixel (x1, y1) reside to the left of the *NCLine* and the *NCLine* (x2, y2), according to the Euclidean distance equations 1-3. Step 3, In most cases, the point with the maximum distance from the *NCLine* considered as the "*Nose Bottom point*". Figure 6(b) shows the horizontal line passing through this point and the red arrow points toward this point.



(c) (d)

Figure 6. Nose Above and Bottom Points: (a) Line Passing from Nose Point to Chin Point; (b) Line Passing through Nose Bottom Point; (c) Nose Above and Nose Bottom Points; (d) Line Passing through Nose Above Line.

4.4 Nose Above Point

This point is the one that resides above the nose point peak and the nostrils reside below it. The following steps are used to detect this point. Step 1, locate the Nose point feature pixel which determined previously. Step 2, draw a circle such that the nose point defined as its center and with a radius defined as the distance from the nose bottom point to the circle center. Step 3, In most cases, the top point of the face skin region touched the circle is considered as the "*Nose Above Point*". Figure 6(c), illustrates the Nose features that will be used in our FD algorithm for colored image. While, Figure 6(d) shows the line passing through the Nose above point in the segmented face image and the red arrow point toward the Nose above point.

4.5 Neck Point

Neck point is one of the features used in this work. The following steps summarize the procedure for detecting this point. Step 1, *a horizontal projection* for the face skin pixels in a face image is taken as shown in Figure 7 (b). Step 2, draw a line passing between the beginning point and maximum point of the previous calculated horizontal projection values calculated in previous step as shown in Figure 7 (c). Step 3, project the segmented skin pixels which are below this line on this line. Step 4, The pixel with the *maximum distance* from this line will be the *Neck Point* as shown in Figure 7 (d).

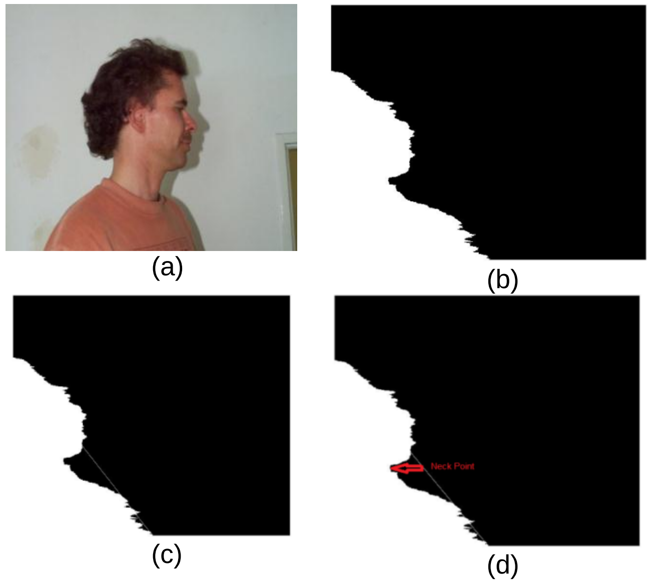


Figure 7. Neck Point Marking Process: (a) Original Image; (b) Image Horizontal Projection; (c) Line Passing between the Beginning Point and Maximum Point of the Projection Histogram; (d) Neck Point in the Projection Histogram.

4.6 Angle Between (Nose, Nose Above and Nose Bottom) Points

The Nose, Nose above and Nose bottom points together compose a triangle and each one of these points represents a vertex of this triangle. In our work we focus on the Nose angle which is the vertex that restricted between Nose Top and Nose Bottom points. We calculate this point and call it ("*NoseAbove_Nose_NoseBottom*"). Figure 8 shows this point marked with red bubble.

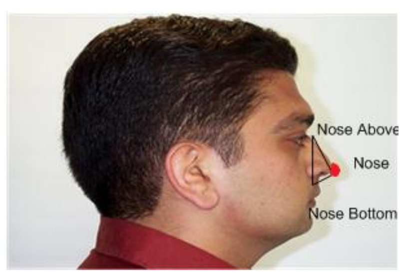


Figure 8. The Main Three Angles in our FD Algorithm

5. The Proposed Feature-Based FD Algorithm

The proposed algorithm evaluates some rules and thresholds that judge if the image indicates a face or not and these rules use the previously calculated features for this judge. We do a lot of experiments studying how we can define these thresholds and rules, and from training and testing we reach that in most cases the following rules are correctly satisfied to judge if these calculated features represent a face or not:

- ✓ **Rule 1:** NoseAbove_Chin_Distance > (Height_Face_Region / 4) .
- ✓ **Rule 2:** Nose_NoseBottom_Distance < (0.5 * Nose_Chin_Distance) .
- ✓ **Rule 3:** (0.9 * Nose_Head_Distance) < Nose_Chin_Distance < (1.3 * Nose_Head_Distance).
- ✓ **Rule 4:** 45 < NoseAbove_Nose_NoseBottom < 150

Where,

NoseAbove_Chin_Distance is the vertical distance between the Nose above feature point and the chin point. *Height_Face_Region* is the height of the face skin region. *Nose_NoseBottom_Distance* is the vertical distance between the Nose point y-index coordinate and the nose bottom x-index coordinate. *Nose_Chin_Distance* is the vertical distance between the Nose point and the Chin point. *Nose_Head_Distance* is the vertical distance between the start of the skin region (head start) and the nose point.

If the calculated face rules applied successfully to the image, this means that the image has a profile face and its coordinates will be (x, y, Width, Height).

where,

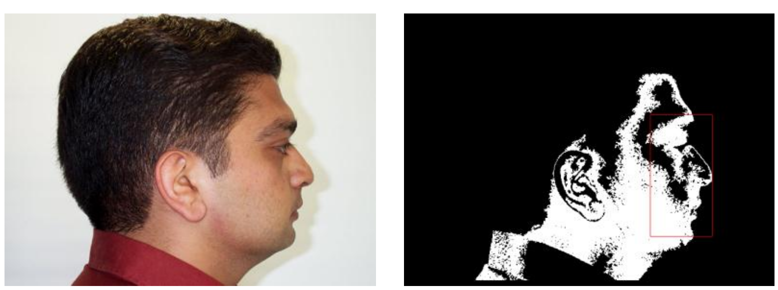
x = (Nose Point y-Index coordinate) – (*Nose_Chin_Distance*) .

y = (Nose Point x-Index coordinate) – (*Nose_Chin_Distance*) .

Width = *Nose_Chin_Distance* .

Height = 2 * (*Nose_Chin_Distance*) .

Figure 9(b), shows the detected face region with a red rectangle on the right profile segmented image.



(a) (b)

Figure 9. FD Algorithm Output: (a) Original Image; (b) Detected Right Profile in Segmented Face

6. Results and Discussion

We have implemented and evaluated our proposed FD algorithm using Matlab v. 7.6, which run under Delll-inspiron laptop with a 2-GHz CPU and has a 3-GB DDR2 Ram. To test our implemented work, we carry out several tests on two different databases. The first is **GTAV** database [10], the second is **FEI** database [11]. These databases contains colorful people images under different illuminations conditions and with different poses and races. The resolution of the images vary from 240x320 to 640x480. These databases include a lot of images with profile rotation of up to about 180^o degrees with scale might vary about 10%. Figure 10, shows the successful results of applying our FD method to GTAV face database. While Figure 11, shows the successful results of applying our FD method to samples of FEI face database. Each one of these figures contains the three face positions which are (Right Profile, Left Profile images).

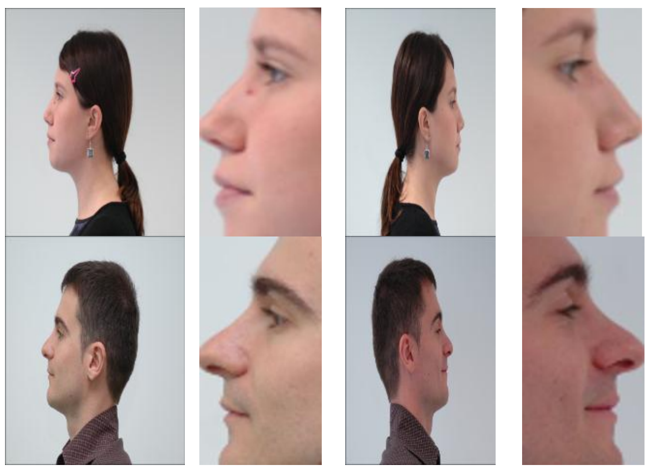

(a) (b) (c) (d)

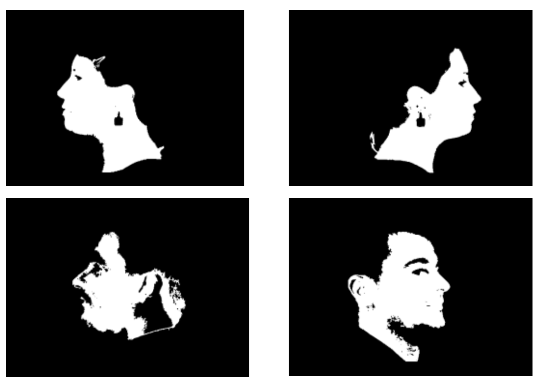
Figure 10. Result Samples of Applying Proposed FD Algorithm for Persons in GTAV Database: (a) Original Left Face; (b) Left Profile; (c) Original Right Face (d) Right Profile.

6.1 Segmentation Process

As known, any FD algorithm contains some pre-processing steps that leads to correct FD process. The most important pre-processing step in FD is *face segmentation*. The segmentation algorithm segments the face correctly from the image then this leads to accurate FD algorithm. we build an efficient face segmentation algorithm and this segmentation algorithm passes through different filtering steps. Figure 12 illustrates the output of our proposed segmentation algorithm applied to samples images from GTAV face database.

(a) (b) (c) (d)

Figure 11. Result Samples of Applying Proposed FD Algorithm for Persons in FEI Database: (a) Original Right Face; (b) Right Profile; (c) Original Left Face; (d) Left Profile.



(a) (b)

Figure 12. Results of Applying Proposed Segmentation Algorithm to GTAV Face Database: (a) Left Profile; (b) Right Profile.

6.2 Proposed Face Detection Algorithm Evaluation Results

In this section, a batch of evaluation have been done to evaluate, among others, the following parameters on different conditions: (1) **Number of false Detection**: How may number of times the proposed algorithm fails in FD process. (2) **Correct Detection Rate**: The ratio of the correctly detected faces.

In the following tables, a detailed performance results for our selected databases when used in our proposed FD algorithms are given. These results partitioned according to the position of the face. In FEI and GTAV face database tests, we use test images with *right*, *left*, face images. For the FEI database, the correct detection rate is 91.5% and, for the GTAV database, it is 93.2%. Tables 1 and 2 show the detail detection results for both.

Table 1. Detail Results of Testing my Proposed FD Algorithm on FEI Face Database

Test \ Test Type	Right profile	Left profile	Total
No. of images	100	100	200
No. of false Detection	10	7	17
Correct Detection Rate	90%	93%	91.5%

Table 2. Detail Results of Testing my Proposed FD Algorithm on GTAV Face Database

Test \ Test Type	Right profile	Left profile	Total
No. of images	260	255	515
No. of false Detection	20	15	35
Correct Detection Rate	92.3%	94.1%	93.2%

6.3 Discussion and Comparison

Our proposed algorithm gives efficient results compared to other approaches. To see how efficient our proposed algorithm work, our results are compared with other relative techniques work in this field. We choose the work proposed by Hsu et.al [9] to compare with. They propose an algorithm for FD that is similar to our algorithm because it detect the face in complex background and with varying lightening conditions. So, we select Hsu et.al. research for our comparison. In this comparison, we evaluate the correct FD percentage parameter and the obtained results are presented in Table 3.

Table 3: Results of Comparing my Proposed Algorithm with other Algorithm

Test \ Algorithms	Proposed Algorithm	Hsu. Algorithm
Correct Percentage %	94.5%	91.63 %

It is shown from Table 5.5 that our algorithm gives better performance than Hsu et.al [9] nearly by 2.87 %, which is an efficient result in this filed.

6.4 Case Of Failure

However the proposed FD algorithm gives a satisfied results in most cases but, there are still some error factors can appears due to different reasons. Figures 13-16, present some of these cases and the conditions which lead to failure in FD process.

- **Case 1: Segmentation Error**

Inspite of the robustness of our proposed face segmentation algorithm but, in some cases the algorithm face some obstacles that lead to failure in detection process. Figure 13, shows an example of failure happened when the person wearing clothes which are very relevant to skin color hence, the proposed algorithm mark these clothes as a skin pixels and this will lead to fail FD process.

In some case, the long hair with the similar skin color leads to fail in FD process. Figure 14, shows an example of this failure.

Fail !

(a) (b) (c)

Figure 13. Clothes with Similar Skin Color Lead to Face Segmentation Error case: (a) Original Image; (b) Segmented Image; (c) Detected Face Image.

Fail !

(a) (b) (c)

Figure 14. Long Hair with Similar Skin Color Lead to Face Segmentation Error case: (a) Original Image; (b) Segmented Image; (c) Detected Face Image.

- **Case 2: Noisy Image**

The noise in the image is one of the obstacles that faces the FD algorithm. Noise may occur in digital images because of many reasons such as camera lens, environment conditions, lightening sources, …, etc. Figure 15, shows an image with some noise which leads to fail in segmentation process.

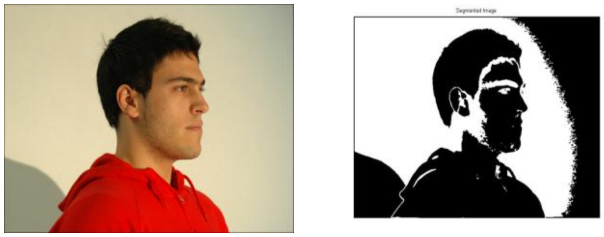
Fail !

(a) (b) (c)

Figure 15. Noise in the Image: (a) Original Image; (b) Segmented Image; (c) Detected Face Image.

- **Case 3: Irregular Illumination Condition**

The illumination conditions are important factors that affect the FD process. As shown in Figure 16, the lightening source is focused in a direction with some shadow that leads to miss-classification in our proposed algorithm.



Fail !

(a) **(b)** **(c)**

Figure 16. Irregular Illumination Conditions: (a) Original Image; (b) Segmented Image; (c) Detected Face Image.

7. Conclusion

In this paper we have proposed a face detection algorithm depending on feature-based approach for detecting face images with profile poses, races and fall under different illumination conditions. This algorithm proposed an efficient segmentation procedure that filter the image from ambiguous objects and determine if the face image contains shoulders or not, and cut this part from the face region if exists depending on a neck point as a reference point. Evaluation results show that the proposed algorithm is robust over different lightening conditions, races and profile posed. The proposed face detection algorithm gives a minimal false detection and high correct detection rate that Hsu et.al [9].

References

[1] W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld, "Face Recognition: A Literature Survey," ACM Computing Surveys, vol. 35, no. 4, **(2003)** December, pp. 399–458.

[2] C. Zhang and Z. Zhang, "A Survey of Recent Advances in Face Detection," Microsoft Corporation, Redmond, WA USA , Technical Rep. MSR-TR-2010-66, **(2010)** June.

[3] T. Sakai, M. Nagao, and T. Kanade, "Computer analysis and classification of photographs of human faces," presented at Proc. of 1st USA-Japan Computer Conf., Kyoto, Japan, **(1972)**, pp. 55-62.

[4] N. Oliver, A. Pentland, and F. Bérard, "LAFTER: A real-time face and lips tracker with facial expression recognition," in Elsevier J. of Pattern Recognition, vol.33, **(2000)**, pp.1369–1382.

[5] E. Hjelmås and B.K. Low, "Face Detection: A Survey", IEEE Trans. Computer Vision and Image Understanding, vol. 83, no. 3, **(2001)** September, pp.236-274.

[6] R. Brunelli and T. Poggio, "Face recognition: Features versus templates", IEEE Trans, On Pattern Analysis and Machine Intelligence, vol.15, no.10, **(1993)**, pp.1042–1052.

[7] T. Kanade, "Picture Processing by Computer Complex and Recognition of Human Faces", PhD thesis, Kyoto University, **(1973)**.

[8] F. Y. Shih and C. F. Chuang, "Automatic extraction of head and face boundaries and facial features," Inf. Sci. vol. 158, no.1, **(2004)** January, pp.117–130.

[9] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K. Jain, "Face Detection In Color Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, **(2002)** May.

[10] F. Tarrés, A. Rama, "GTAV Face Database," available: http://gps-tsc.upc.es/GTAV/ResearchAreas/UPCFaceDatabase/GTAVFaceDatabase.htm.

[11] "FEI face Database", **(2006)** March, available: http://fei.edu.br/~cet/ facedatabase.html.

Authors

Shady S. Al-atrash

Graduated in bachelor's degree in Computer Engineering from the Islamic University of Gaza in 2007 and received his M.Sc. degree from the Islamic University of Gaza in 2001. Eng. Shady is now instructor at the Islamic University of Gaza. His research interests are on computer vision subjects especially Image Processing, Machine Learning and Artificial Intelligence.

Ibrahim S. I. AbuHaiba

Graduated in bachelor's degree in Science from the Al-Azhar University, Cairo, Egypt in 1984 and received his M.Sc. degree in Philosophy from the University of Bradford, Bradford, U.K in 1991. Received his Doctora degree from Loughborough University, Loughborough, U.K in 1996. prof. Ibrahim is now instructor at the Department of Electrical and Computer Engineering in Islamic University of Gaza. His research interests are on Pattern Recognition, Machine Learning, Artificial Intelligence, Document Analysis and Understanding, Optical Character Recognition, Computer Vision, Digital Image Processing, Robot Modeling and Control, Data Mining, Optimization Techniques, Advanced Embedded Systems, Real-time Systems, Wireless Networks, Computer and Network Security, Fault Tolerance and Reliability in Computer Networks and Software Development.