Component-Based Real Time Facial Expression Recognition in Video Streams

Sara Alipour and Afsane Fathi

Abstract—**In this article a new real-time approach is proposed to recognize facial expression in video database. First, face motion direction is estimated by using motion information and then according to the face width and direction, rectangular surrounding box is formed. The advantage of this method compared with previous methods is in improving classic rectangular surrounding box model and also, using the Steerable filters to determine face direction. As eyes and mouth are the most important components in a face, in this method, eye location is calculated by combining color features and morphological functions in surrounding box region. Afterwards, with regard to the lip contour and using image processing techniques, we are able to determine the detection of mouth very accurately with the help of four points. Finally, these features are modeled by using support vector machines (SVM). Experimental results show that this algorithm is more efficient than the previous ones.**

Index Terms—**Facial expression recognition, rectangular surrounding model, support vector machines.**

I. Introduction

Tracking the human face is one of the interesting and important topics to discuss in machine vision issue which is used to recognize the face and facial expressions and also to identify speech. Face detection in the image is the first step in any facial processing system [1], [2]. This step could be used as a preprocessing step to apply to higher-level processes such as facial expression recognition which is the aim of this article. The purpose of facial expression recognition is revealing the position of face components such as eyes, eyebrows and mouth [3].

In the past two decades, much research has been done in this domain. Particularly from 1994 onward, varieties of methods have been used to track the human face. Among these, algorithms are considered more important which compute faster. This paper is proposed a real time method with the highest possible accuracy. A real-time method based on rectangular surrounding box model to detect facial expression inscribed in [4], which idea is presented by Viola in [5] that is able to detect any size of face in the image as a real-time. In this technique a series of features with rectangular structure is used for instruction.

Continuously, a new algorithm is recommended that first tries to determine the face motion direction and find the exact range of the face according to rectangular surrounding model.

Manuscript received January 15, 2013; revised May 16, 2013.

Sara Alipour is with the Young Researchers Club, Islamic Azad University, Qazvin Branch, Qazvin, Iran (e-mail: alipour62@ yahoo.com).

Afsane Fathi is with the Islamic Azad University, Qazvin, Iran (e-mail: Afsane.fathi@hotmail.com).

Consequently, a brief description of classical rectangular surrounding method is expressed and according to its defects, the new algorithm will be presented. Not only it keeps real-time response time but also it has higher accuracy than the former algorithm. Also, the method is provided to detect eyes and eyebrows by using color features and morphological functions. Eventually, with the help of four points the location of mouth is determined and all obtained features are modeled by support vector machines (SVM) to extract facial expressions.

This paper is structured as follows. In Section II, we concentrate on how the face direction estimates are obtained. Section III deals with face boundary detection on video datasets by challenging classic model. In Section IV, V respectively, present the eyes and mouth detection model. Experiments are accomplished on XM2VTS database in comparison traditional method in Section VI. Finally，we discuss our results and offer some conclusions.

II. Detecting Face Motion Direction

To determine face motion direction is used from the dominant image direction. Following this concept, local purpose is finding blur or tremor motion in the image. For this intention can use direct linear filters (Steerable Filters).

A. Steerable Filters

These filters are in order to find the motion blur in the image. Interactively show maximum response from self-evident of blur motion and be able to calculate that the dominant image direction in an image [6]. Filter is obtained by any combination of linear filters for a particular base, and therefore only we calculated response a limited number of basic filters and output of other angles is calculated with a simple linear combination.

Filter in θ direction is calculated by (1):

$$f^{\theta}(x,y)=\sum_{j}k_{j}(\theta)f^{\theta_j}(x,y). \quad (1)$$

$$k_j(\theta)=\frac{1}{3}[1+2\cos(2(\theta-\theta_j))]. \quad (2)$$

In which $\theta_j=\frac{j\pi}{3}$, $j=0,1,2$, f^{θ_j} is the based filter on θ_j direction. For the final motion is calculated in (3) and variables f_e^{θ} , f_o^{θ} are calculated in (4) and (5):

$$\theta_b=\arg\max_{\theta}((f_e^{\theta}(x,y))^2+(f_o^{\theta}(x,y))^2). \quad (3)$$

DOI: 10.7763/IJMLC.2013.V3.328

$$f_e^{\theta}(x,y) = 0.9213(2x^2-1)e^{-(x^2+y^2)}. \quad (4)$$

$$f_o^{\theta}(x,y) = (-2.205x+0.9780x^3)e^{-(x^2+y^2)}. \quad (5)$$

From this angle, we can be estimated face motion direction and can be used in the proposed algorithm.

III. Detecting Face Boundary

It should be noted that a variety of methods is available to determine the boundary of face such as Oval border Model [7], Hexagonal Model [8] and Rectangular Surrounding Box Model [4]. In this paper, due to the ease of calculations and appropriate accuracy is discussed Rectangular Surrounding Model.

A. Classical Method of Rectangular Surrounding Box Model

This algorithm tries to extract the face surrounding rectangular coordinates of vertices in video images. For this reason first, all frames are converted to gray levels. After that according to (6) the difference in brightness between frames D is calculated.

$$D_k(i,j) = \left| I_k(i,j) - I_{k+1}(i,j) \right| \\ 1 \le i \le w, 1 \le j \le h, 1 \le k \le m \quad (6)$$

In (7) *I* variable means gray level image that is associated with each frame. Then threshold value applies to obtained image and *T* image is obtained that contains the appropriate motion information. After getting the threshold image, review to find the highest stimulus point (c_x, c_y) is performed. Also the highest stimulus point to review threshold image due to find head contour points. This process is done that head contour points separately will find over from the left contour point (l_x, l_y) and the highest stimulus point and also, from right contour point (r_x, r_y) and the highest stimulus point.

With calculating these points we can achieve the coordinate the left, right and above sides of the rectangular surrounding. And then the underside of the box is placed $4/3(r_x–l_y)$ lower than the upper side of the box [4].

B. Weaknesses of the Classical Algorithm

One of the major defects of the classical algorithm is related to find the appropriate highest stimulus point. As seen in Fig. 1, the highest stimulus point is placed in its proper place only, when threshold value is about 50 and also, many parts of the face contour might be removed by this threshold value. If the highest stimulus point is not choose the right place, find a good selection leftmost or rightmost contour point as possible and cause that will not obtain appropriate rectangle. Even with significant movement if the distance between the leftmost and rightmost point is not obtained larger than $4/3(r_x–l_y)$, it leads rectangular surrounded location is not obtained properly.

Another important defect of this algorithm is related to extract the leftmost and rightmost contour point. Fig. 2 shown that even to find a proper place for the highest stimulus point, again algorithm may not work as well to find the most appropriate left and right contour point.

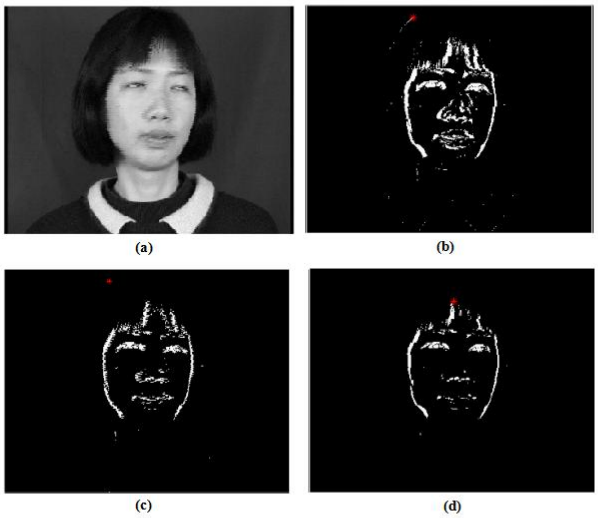

Fig. 1. (a) Moving face to the right side, the threshold value in the (b), (c) and (d) images, is respectively 30, 40 and 50. Red point displays where the highest point of stimulus.

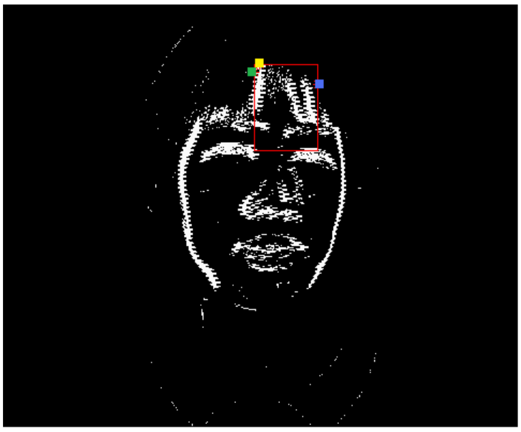

Fig. 2. Points of yellow, blue and red are respectively the highest stimulus point, and red rectangular box is face rectangular surrounded.

C. Proposed Algorithm

We can classify the proposed algorithm as follows:

First, we find the highest point with non-zero brightness level in the threshold image, Afterwards we can estimate the direction of face by using Steerable filters. In that approach, two different positions are considered.

If the picture moves toward left or right for each side of the of the face the highest moving points (c_x, c_y) are calculated. for example for the right side when the face moves toward left, first the highest moving point should be negative and secondly from right to left we look for the highest moving point. This point will be the highest moving point for the right side which is used to find out contour points of the right side of the face. Such a work is done to find out the highest moving point for the left side with this difference that we start from the left side to right and know that this time the highest point should be positive. After finding the most left and the most right points of contour points of the face from the two points, the one that is highest, will be selected.

If the picture moves up or down, we find angles of square only, from contour points. But we should know that pixels value when the face moves up is positive and it is negative when the face moves down. In this position with the previous classic method, we look for the highest moving point. The important point is in each part, the highest moving point should have 15 neighboring points with same sign in 7×7 distances from itself. This limitation is considered to avoid choosing isolated points to find out the highest moving point.

Before finding contour points closing the picture on brightness difference picture is done. Constructive elements is used horizontal when movement is horizontal and vice versa. The point to consider is that, the length of constructive element for negative pixels is more than positive pixels.

Then by having the highest moving points, we look for contour points of the face. We should know that when the face moves horizontally to find out contour points of the left side (l_x, l_y) and right side (c_x, c_y) will have an isolated highest moving point. the contour points of picture is shown when movement is from the highest moving point to the most left point based on the suggested algorithm is shown in Fig. 3.

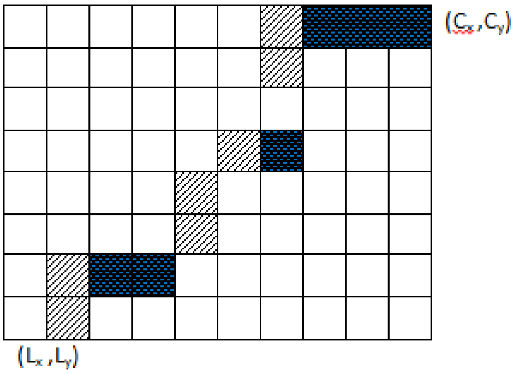

Fig. 3. Bright areas are parts of the image contour.

After finding the main contour points of the face, Angles of square which covers the face are found as (7). In which (r_x , r_y) is the width of the face.

$$(l_x, c_y), (r_x, c_y), (l_x, c_y + \tfrac{4}{3}(r_x - l_x)), \quad (r_x, c_y + \tfrac{4}{3}(r_x - l_x)). \tag{7}$$

If the number of contour face is less than the width of the face found out the same previous square is again considered. This limitation is considered to have appropriate movement to change the square which increases the accuracy of the system.

IV. Detecting Eyes and Eyebrows Locations

In this step, first YC_rC_b is found out by using the colour features from RGB then, obtained number are normalized in to the (0, 225) zone. As eye has low intensity (y), low red colour C_r , high blue colour C_b than forehead so we found out the U map[8]- [10].

In most of the algorithms to delete empty points in a picture, we use closing of the picture [11], [12], but to delete continuity of the pictures first opening is done and then closing .which is more useful to find the location of the eyes. Fig. 4 shows the progress of eyes detection from original image.

After opening and closing operation, we found the blobs, which are the eyes in the image. Each blobs that their length or height are more than 0.25*W1* are not considered.

W1 is the width of the image. Centred of the remaining blobs are calculated to find out the two points around the eyes.

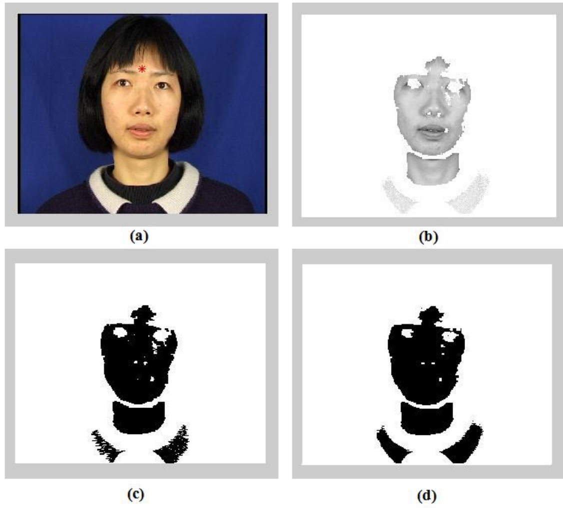

Fig. 4. (a) Original image, (b) U map from the (a) image, (c) output from the closed operation and, (d) Output from opening and closing of the image. Closed operation done on image to all structural elements 5×5 array. The red star shows the forehead location.

The data for eyebrows are found out as (8):
Previously, values of λ were calculated.

$$E(i,j) = \begin{cases} 1 & \text{if} \quad Y(i,j), Y(i,j+1) \geq \lambda_1 \;,\; Y(i,j+2) < \lambda_1 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

V. Mouth Detection

Mouth area in colour image consists of strong red and weak blue in comparison to other points. So that in chromatic space C_r should be bigger than C_b. After finding the location of mouth zone, the exact location of mouth is found out [13]. In these way four points of mouth are found higher point of upper lip, lower point of lower lip and two points in left and right.

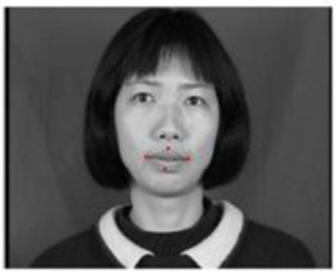

Fig. 5. Two points of above and below the lip position on vertical axis of mouth found between two corners of the mouth.

This way by using maximum negative of the right side and left side of the horizontal histogram, found out from horizontal lines of the square for the highest and lowest points on lips will be found out . At last, we use lips detection to find out the two ends of lips. Fig. 5 shown sample of our mouth detection.

VI. Experimental Result

SVM has been used for modelling in order to identify patterns. For data which can be separated linearly, hyper plane SVM separated this class of models [14]. For data which cannot the separated linearly, SVM writes the data in higher spatial dimension and finds a place which can separate the two classes.

We use M2VTS data base for test this algorithm, which has different light condition and has head movement in four main directions. For recognize the motion direction, filter with 14×14 pixels has been applied. Characteristics vector has been calculated of 13 different features (four points for the lip, four points for the two eyes, four points for the two eyebrows and one for forehead). To test 20 features vector, 20 frames of SVM were used. The Experimental results show that a higher accuracy than Previous methods. In face boundary detection with comparison to algorithm [4] accuracy improved and it reached 98.6%. Both algorithms were investigated on the computer with 3.0 GHz CPU as 10 frames per second. In general, detection rate is 96.4% and also, by applying proposed reformation the reforms being real-time specifications is well maintained property.

However, in both cases, we assume that another moving object except the head of a human is not present in the image. In Fig. 6 a sample of face Rectangular Surrounded box by using method available in [4] and the method proposed in this paper has been compared.

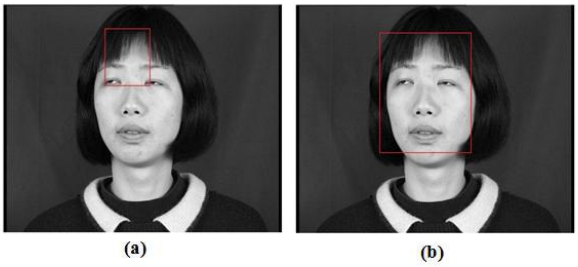

Fig. 6. Finding face rectangular surrounded while face is moving to the left. (a) using the algorithm [1] (b) using the proposed algorithm.

VII. Conclusion

The aim of this study is to present a new (no time delay) method with high efficiency to recognize different facial expressions in a video streams which is robust in different light conditions and face movements in all directions up, downright, left. Therefore, in the proposed algorithm first, we find out the direction of movement in filters and, then by presenting a new rectangular surrounding box which is more accurate than the previous classical model, the location of the face is computed. Also, by combining colour features and morphological functions the location of eyes and finally with regard to the lip contour the location of mouth is identified. After identifying the location of mouth by SVM, all features are extracted based on the facial expressions. Clarifying value of 96.4% is found which is more accurate in comparison with the previous models.

References

[1] N. Esau, E. Wetzel, L. Kleinjohann, and B. Kleinjohann, "Real-time facial expression recognition using a fuzzy emotion model," in *Proc. Fuzzy Systems Conf.*, 2007, pp. 1-6.

[2] I. Cohen, N. Sebe, A. Garg, M. S. Lew, and T. S. Huang, "Facial expression recognition from video sequences: temporal and static modeling," *Computer Vision and Image Understanding*, vol. 91, no. 1-2, July 2003.

[3] S. M. Bileschi and B. Heisele, "Advances in component based face detection," in *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003, pp. 149–156.

[4] A. Geetha, V. Ramalingam, S. Palanivel, and B. Palaniappan, "Facial expression recognition–a real time approach," *Expert Systems with Applications,* vol. 36, no. 1, pp. 303–308, 2009.

[5] P. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2001, pp. 511-518.

[6] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 13, no. 9, pp. 891-906, 1991.

[7] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 1, pp. 274-280, 1999.

[8] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, 2002.

[9] K. M. Lam and H. Yan, "Locating and extracting the eye in human face images," *Pattern Recognition*, vol. 29, no. 5, pp. 771–779, 1996.

[10] F. Smeralsi, O. Carmona, and J. Bigun, " Saccadic search with Gabor features applied to eye detection and real-time head tracking," *Image and Vision Computing*, vol. 18, no. 4, pp. 323–329, 2000.

[11] P. T. Jackway and M. Deriche, "Scale-space properties of the multiscale morphological dilation-erosion," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 38–51, 1996.

[12] A. Nikolaidis and I. Pitas, "Facial feature extraction and pose determination," *Pattern Recognition*, vol. 33, no. 11, pp. 1783–1791, 2000.

[13] M. Malciu and F. Preteux, "Tracking facial features in video sequences using a deformable model-based approach," *SPIE*, vol. 412, pp. 51-62, 2000.

[14] H. C. Lu, Y. J. Huang, Y. W. Chen, and D. I. Yang, "Real-time facial expression recognition based on pixel-pattern-based texture feature," *Electronics Letters*, vol. 43, no. 17, pp. 916–918, 2007.

S. Alipour was born on August 4, 1983. She received the bachelor's degree in Computer Software Engineering from Behshahr University in 2006. She has been a student in the field of Artificial Intelligence in master's degree at Azad University of Qazvin (QIAU) since 2009. She is a member of Young Researchers Club, Islamic Azad University, Qazvin Branch, Qazvin, Iran. Her main research interests are in the fields of image processing, image analysis, computer vision, pattern recognition. She participated to a number of national and international research projects facing automatic video surveillance, face detection, real time event detection in sport contexts. She is the author of some papers in national and international conference proceedings.

A. Fathi was born in Tehran, Iran in 1985. She got her B.Sc. from Islamic Azad University, Arak, Iran in 2007 in Hardware of Computer Engineering. She was graduated M.Sc. degree of Computer Engineering field of Artificial Intelligence at the Islamic Azad University, Qazvin, Iran in 2012. Her research interests are image processing, pattern recognition and neural networks.