Face Recognition Technology based on Kalman Filter

Lili Wang

School of Information Engineering, Harbin University, Harbin 150001, China

108587570@qq.com

Keywords: Lip movement; Face recognition; Pattern recognition; Computer vision; Human computer interaction

Abstract. In this paper, a combination of basic Haarlike features proposed by Viola-Jones an extended set of Haarlike features proposed by Lienhart et al will be researched and implemented in the face detection phase using matlab and a set of proposed rules for Adaptive Boosting(Adaboost). Followed by a method of Lip Localization using YCbCr colourspace, lip contour estimation with the Van-Chese method of active contours without edges and tracking using kalman filter algorithm. The significance of the item is that it can be applied to human-computer interaction, security and surveillance, video communication and compression, medical imaging, video editing and robot vision applications. This paper can also serve as groundwork for future research into automatic lip reading systems which may allow it possible for an application to know what one is saying in a video simply by reading their lips.

Introduction

Detection as the first step is necessary for any image processing methods that require working with the face [1,2]. In the past 20 years, research in the fields of computer vision, pattern recognition and machine learning has attracted researchers all over the world. Face recognition and other video coding techniques have recently received a great amount of attention from researchers. Various research methods have been proposed and implemented particularly in the areas of digital image analysis which spans especially through face detection, recognition and other related areas [3]. Most of these proposed methods assume during experiments that the images are in frontal views and under controlled conditions such as background and lighting which are not true in a real case scenario.

For instance in the case of this item, after face detection mouth localization and must be done so that further estimation of the lip contours can be easily achievable. Different method of facial features have also been proposed and tested and shall be discussed in the sections that follow [4,5,6]. Video based processing has also become a very essential field for computer vision, pattern recognition and machine learning among researchers and video based applications are increasingly in demand [7,8].

This feature assumes that most of the information is contained in contours or shape of speaker's lips. Geometric features such as height, width, perimeter of mouth, can be easily extracted from the ROI [9,10]. Model-based features are obtained in conjunction with parametric or statistical feature extraction algorithm.

Related Research Theories of Methodology

Extended Haarlike Features. The use of features makes classification easier compared to using raw pixel values as input to an algorithm. Haarlike features were first used by Papageorgiou et al and Viola et al later proposed a fast computation method for generalizing their work. A total of fifteen(15) over complete Haarlike feature prototypes proposed by Lienhart et al were looked at and four(4) namely line, edge and the special diagonal feature were used in this implementation. Figure1 shows all the features used. The number of features in a 19 x 19 window sizes since all images obtained from the MIT CBCL are all 19 x 19 pixels in size is over 53000.

  DOI: 10.25236/icmcs.2019.003

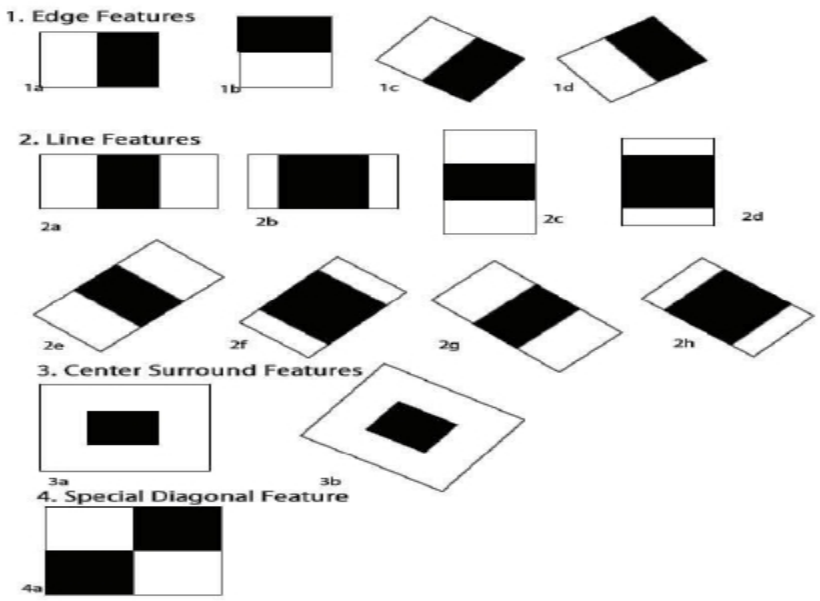

Figure 1 Feature prototypes of simple Haar-like and center-surround features. Black areas have negative and white areas positive weights

The Calculation of the number of features for all upright(0^0) prototypes was determined by the algorithm given in pseudo code:

Algorithm 1. Number Of Features(0^0)

For all pixels inside the boundaries of our feature given (x_i, y_i)
for each width and length possible in frame size(19x19)(X,Y)

Number Of Features = $XY(x_i + 1 - w \times (X+1)/2)(y_i + 1 - h \times (y+1)/2)$

The Calculation of the number of features for all rotated(450) prototypes was determined by the algorithm :

Algorithm 2. Number Of Features

For all pixels inside the boundaries of our feature given (x_i, y_i)
for each width and length possible in frame size(19x19)(X,Y)

Number Of Features = $XY(x_i + 1 - z \times (X+1)/2)(y_i + 1 - z \times (y+1)/2)$

Integral Image. In order to save computational costs, the Summed Area Table(SAT) of each image can be used to determine the number of features in any image of any size. Integral Image (I) and Summed Area Table are sometimes used interchangeably.

Summed area table. SAT (i, j) gives the sum of the pixels of the original image, I within the upright rectangle ranging from the top left corner to the bottom right corner with a single pass over any given image at $I(i, j)$, as :

$$SAT(i,j) = \sum_{x \le i, y \le j} I(i,j) \qquad (1)$$

This relationship between I and SAT is graphically demonstrated in Figure2 .The power of SAT comes from the fact that any summation of the pixels from the original image I within a rectangle could be done by 4 table lookups and 3 addition/subtraction operations in SAT, which is shown in Figure2.

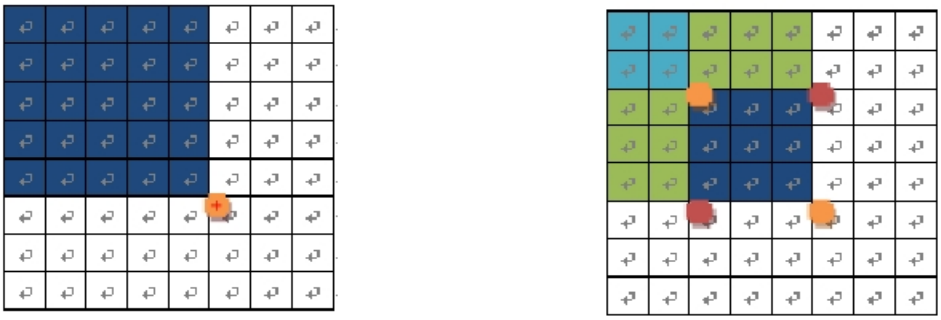

Figure 2 Graphical representation of SAT

Rotated Summed Area Table. RSAT (i, j) gives the sum of the pixels of the original image, I within the rotated (450) rectangle ranging from the right most corner through all boundaries with also with a single over any given image at $\mathrm{I}(i, j)$, as:

$$SAT(i, j) = \sum_{x-1 \le j-y, y \le j} I(i, j) \tag{2}$$

The relationship between I and RSAT is graphically demonstrated in Figure3 Similarly, any summation of the pixels from the original image I within a 450 rotated rectangle could be done by 4 table-lookups and 3 basic binary operations in RSAT as in Figure3.

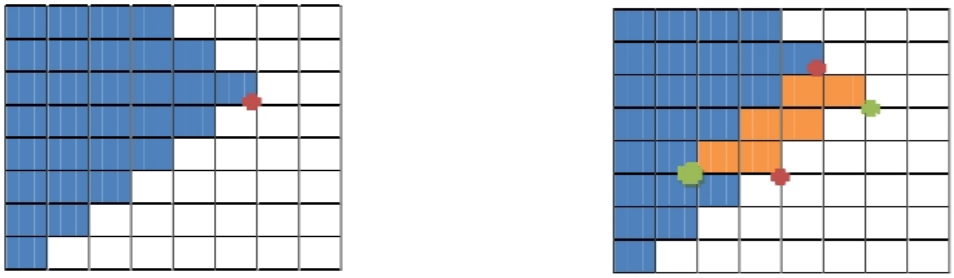

Figure 3 Graphical representation of RSAT

Active Contours Without Edges (Chan-Vese Model). The snake model was very successful and variations of it were adopted later on by several researchers. Most of these models use the level set formulation for propagating fronts to evolve the curve. But these models highly depend on motion defined by the gradient of the image which leads to poor performance in images without gradient and therefore, application of the classical snake might be a very difficult task. To solve these limitations, Chan and Vese proposed an active contour model that does not depend on the edges (the gradient).They used a region-based approach based on the Mumford-Shah model and Osher-Sethian level sets to divide the image into two regions: one inside the propagating curve, and one outside. The curve is at the boundary of the object if there is no difference in intensities both inside the curve and outside the curve. This algorithm is not based on edge function to stop evolution curve as desired edge. Chan-Vese algorithm can detect boundary of object which is not defined by gradient, while classical active contours can't be applied. This method also can detect any object by specifying an initial curve in the image, not necessarily surrounding the object. The objective of Chan-Vese model is to partition an input scalar image into two possibly disconnected regions: foreground and background of low intra-region variance and separated by a smooth closed contour .It is represented by the equation:

$$F(C, c_1, c_2) = \mu L(C) + \nu A(\Omega) \tag{3}$$

The Kalman Filter. The Kalman filter is a set of mathematical equations that provides an efficient recursive mean for estimation of the state of a discrete process, in a way that

minimizes the mean of the squared error. The filter performs the state vector estimation in two phases: prediction and correction or updating. The predict step predicts the current location of the moving object based on previous observations. For instance, if an object is moving with constant acceleration, we can predict its current location a_k, based on its previous location a_{k-1}, using the equations of motion. The update step takes the measurement of the object's current location (if available), z_k, and combines this with the predicted current location, a_k, to obtain an a posteriori estimated current location of the object, a_k. The main equations that of the Kalman filter are shown in Figure4.

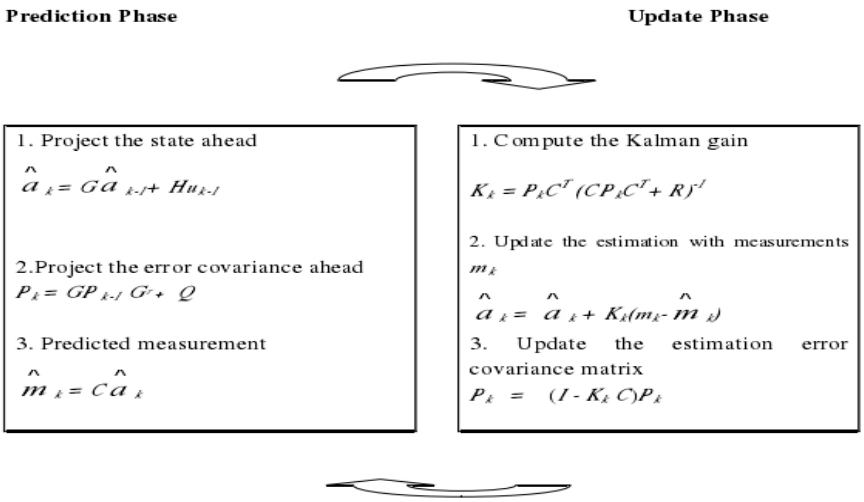

Figure 4 Equations of the Kalman Filter

Implementation of the method

Face Detection. Our own version of The Viola Jones Face Detection System was trained and implemented using Matlab.

Preprocessing. The Dataset used was the MIT CBCL Database downloaded from: http://cbcl.mit/software-datasets/Face Data2.html with description as follows:

1.19 x 19 Grayscale PGM format images

2. Training set: 2,429 faces, 4,548 non-faces

3. Test set: 472 faces, 23,573 non-faces

All the images were normalized to have a unit variance and zero means. This is done to scale the pixel intensity (brightness values) of the images so that dark areas become darker and bright areas become brighter.

Training. For experimental purposes, two sets of training using the boosting method proposed in[1]. The first set contained all fifteen feature prototypes which included the extended haar-like features. The second set of training contained only four feature prototypes which include the two-rectangle features. For the first training set, a total of over 50000 features were obtained for a 19X19 window size and for the second training set, a total of over 36000 features were obtained.

Tracking (Kalman Filters). Multiple Kalman filters were used in this item for motion tracking of the detected faces. The tracking was based solely on the movement of each detected face. Determining the movement of the faces was based on background subtraction based on Gaussian mixture models which is easily achievable in matlab. Finally, blob analysis detects

groups of connected pixels, which are likely to correspond to moving objects. Tracking was implemented with the following steps:

Setp1: Initializing Tracks by creating an array of tracks of each moving object. Each face detected in the video is assigned a new track. This is done to maintain the state of a tracked object for assignment, termination and display

Setp2:Blobs and bounding boxes around the detected face are returned with a mask that has foreground detection set to 1 and background detection set to 0.

Setp3:Kalman filter was used to predict the centroid of each track in the current frame, and update its bounding box accordingly.

Setp4: Compute the cost of assigning every detection to each track. The cost takes into account the Euclidean distance between the predicted centroid of the track and the centroid of the detection. It also includes the confidence of the prediction, which is maintained by the Kalman filter. The results are stored in a MxN matrix, where M is the number of tracks, and N is the number of detections. Then solve the assignment problem represented by the cost matrix

Setp5: Update each assigned track with the corresponding face detection

Setp6: Create new tracks from unassigned detection. Assume that any unassigned detection is a start of a new track.

Setp7: Tracking was done with the assumption that all objects move in a straight line at a constant.

Experiment Design and Discussion

Face Detection. For testing the performance of the face detection system a dataset of faces in captured images were run through the face detector. The dataset was the Bao Face Database downloaded from http://www.facedetection.com/facedetection/datasets.htm. It contained two directories one being of captured images of single face and the other, the captured images of multiple faces. The images have been captured under varying conditions including lighting and pose and are of different image qualities. The result is shown in Table-1, the face detection system performs very well with frontal images and under various lighting conditions but performs poorly with side faces. It is shown in Figure5 and Figure6.

Table 1 Detection results from both directories

Bao Dataset Directory	Number Of Images	Time For An Image (Seconds)	Number of Correctly Detected Faces	Number of Wrongly Detected Faces	Number of Undetected Faces
Single Faces	149	3.76	84	40	25
Multiple Faces	221	7.70	150	44	27

Figure 5 Samples of undetected Faces from the single Face Directory

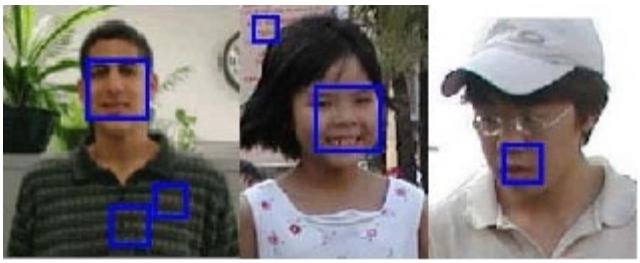

Figure 6 Samples of wrongly detected Faces from the single Face Directory

Mouth Localization and Lip Contour Estimation. Mouth localization was done on images that have already been correctly detected faces. Some of the mouths on the detected faces were not correctly detected. Some faces had too many red and blue components making it difficult to determine the exact location of the mouth since YCb Cr is very sensitive to red and blue. Another factor that was not considered was that the lower lips have higher detection rates since the mouth region has more red components on the lower lip than the upper lip. An adjustment was done such that the position of the bounding box was moved such that for the image bounding box position p(x, y) , p(y) was subtracted by a constant which was determined experimentally. The mouth region for both lip sides is necessary for the effect of lip contour estimation using snakes Figure7 and Figure8 show sample results from the mouth localization using the YCb Cr method

Figure 7 Adjustment made to solve the upper lip detection problem

Figure 8 Partly defined lip boundary

The lip contour estimation proved to be quite a daunting task as the active contour method used as proposed since it was easier to determine the contours of an open mouth as compared to a closed mouth. Values for initial iterations had to be adjusted before the program is run. This is not suitable since it is intended to be an automatic real time segmentation system. A constant value was set at for initialization such that for each detected mouth region (x, y, imagewidth, imageheight) , image iterations begin at index (10,10,imagewidth,imageheight) where (10,10) are values for (x,y)pixels for initialization of the segmentation ,which is suitable for most detections.

The Kalman Filter Results. Since the major objective of this item to track faces with the lip region of videos via webcam, tests were also made for the face and lip detections. Figure9 and Figure10 show tracking results for the implementation of both detections and tracking. The resultant reading of the video frames was a bit slower than expected and there was also some missed face and mouth detection which solely depended on the speed of the face detection system. There was also the issue of jumpy mouth detection as a result of the eyes of some candidates being redder than the lips.

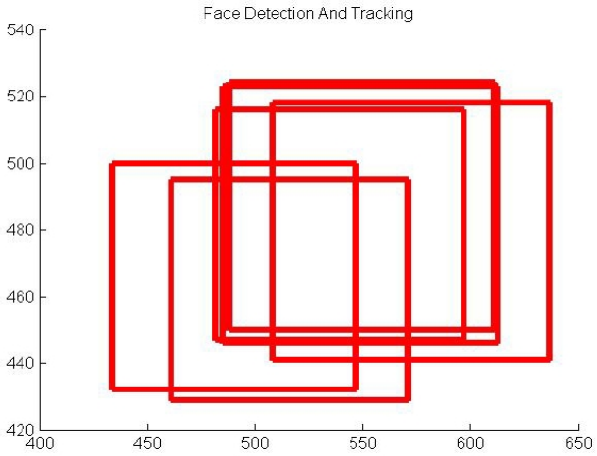

Figure 9 Plot showing Tracking of a Single Face

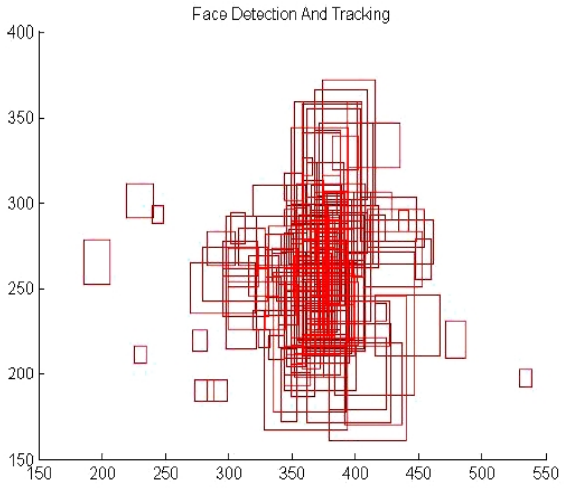

Figure 10 Plot showing tracking results of the mouth

Conclusions

The main objective of the item was to combine different forms of image processing techniques in an effort to develop an efficient system for speaker identification in real time. From the results shown in the previous section, four complete forms of algorithms were tested in the experiments carried out with a few problems encountered:

1. Face Detection results were impressive given that all subjects look directly into webcam. In other words, the system performed really well with frontal face and also under varying lighting conditions but did performed poorly with faces of different poses and side faces.

2 Mouth localization was quite an easy task given that the region for localization was reduced down to the face area. One major problem was that since YCbCr color analysis was used to determine the mouth region, images with several red and blue colour components make it difficult to

achieve a good mouth localization system. This is a necessary step for estimating the contour of the lips so there's a problem of the lip contours not being accurate as a result.

3 Lip contour estimation was also achieved for images with the right color components less blue and red color components with a few difficulties given that the only region for estimation was the mouth area.

4 Tracking of both single and multiple faces are also achieved by using Kalman filters with the initialization of a new track with every detection in each frame.

Several changes can be made in this system to improve its performance. An efficient technique could be proposed to cater for faces under different poses and orientations. A better method can be used for the method of lip localization or different adjustment or adaptive filters can be utilized to solve the problem of too many red and blue components.

References

[1] Luo Lihong, Liu Chunxiao, Keling. Heterogeneous face recognition model based on binary multi-layer GELM. Modern electronic technology, 41(23), 38-43,2018

[2] Duan Hongyan, He Wensi, Li Shijie. Improved face recognition based on single-scale Retinex and LBP. Computer Engineering and Application, 54(23),144-149,2018

[3] Li Qiuzhen, Luan Chaoyang, Wang Shuangxi. Face image quality evaluation based on convolution neural network. Computer application: 12(2), 12-16,2018

[4] Zhong Xiaoli. ASM pose correction combined with dictionary learning optimization for face recognition. Computer engineering and design, 39(11), 3538-3543,2018

[5] Chen Hui. Research on face recognition algorithm based on image block sparse representation. Journal of Xi'an Academy of Arts and Sciences (Natural Science Edition), 21(6), 27-32, 2018

[6] Li Hualing, Wang Zhi, Huang Yujing. Face recognition method based on image features and face posture. Science and technology and engineering, 18(31),195-199,2018

[7] Leung, Shu-Hung; Wang, Shi-Lin; Lau, Wing-Hong; , "Lip image segmentation using fuzzy clustering incorporating an elliptic shape function," , IEEE Transactions on Image Processing , 13(1),51-62, 2004

[8] Kass, M., Witkin, A., and Terzopoulos, D., “Snakes: Active contour models”, International Journal of Computer Vision, 321–331, 1987

[9] Active Contours Without Edges Tony F. Chan, Member, IEEE, and Luminita A. Vese, IEEE Transactions on Image Processing, 10(2), 266-277, 2012

[10] Tang, Z., Miao, Z., “Fast Background Subtraction and Shadow Elimination Using Improved Gaussian Mixture Model,”IEEE International Workshop on Haptic, Audio, and Visual Environments and Games, 2007