

Research of Face Detection Based on AdaBoost and ASM

Han Bing*, Han Xianfeng and Hou Ruizhen

Department of Computer, North China University of Technology, Beijing, China, 100144, P.R. China

Abstract: Face Detection is an essential first step of the face recognition process, and has significant effects on face feature extraction and the effects of face recognition. Face detection has extensive research value and significance. This paper deeply researches and analysis the principle, merits and demerits of the classic AdaBoost face detection algorithm and ASM algorithm based on point distribution model, using ASM to solve the problems of face detection based on AdaBoost. Firstly, the method uses the AdaBoost algorithm to detect original face from images or video stream. Secondly it uses ASM algorithm converges, which fit face region detected by Adaboost to detect faces more accurately. Finally, it cuts out the specified size of the facial region on the basis of the positioning coordinates of eyes. The experiment results indicate that this method can detect face rapidly and precisely, with a strong robustness.

Keywords: AdaBoost, ASM, eye coordinate, face detection.

1. INTRODUCTION

In recent years, Computer face recognition technology has becomes one of the hottest research fields of computer and biometric identification. Face recognition is a process, in which faces are detected from images and video streams, after which those facial features are extracted and the face is identified. The common basic framework of the face recognition system includes images capture, face detection, images pre-processing, feature extraction and face recognition. Although, face detection is one of the key steps in facial recognition, the accuracy of detection has a direct effect on the other stages of the face recognition system [1]. Therefore, how to detect a face accurately and efficiently from images or video stream and how to locate the human face and separate from the background becomes a key issue.

The paper researches and analyses the principle, and the merits and demerits of the classic AdaBoost face detection algorithm and active shape model algorithm based on the point distribution model. Deploying AdaBoost algorithm using ASM solves the problems of face detection. Using AdaBoost improves the accuracy rate of the process and lastly it cuts the facial region according to the eye coordinates positioned by ASM model.

2. FACE DETECTION ALGORITHM

Face detection is a process, which determines whether there are face regions in a particular input image Or video stream and determines the size, location and posture of the face region if it exists. The algorithms commonly used in this process are template matching, SVM and AdaBoost algorithm.

2.1. Face Detection based on Template

Template matching algorithms is of the most basic and commonly used methods of pattern recognition methods. The basic idea is calculating the degree of similarity between the test sample and the standard face sample template and determining whether the sample contains face region based on the relationship between the calculated results above and the specified threshold and identifying the matching position. The equation as follows

$$r(T, R) \approx \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (T[i][j] - \mu_0)(R[i][j] - \mu_0)}{M \cdot N \cdot \sigma_0 \cdot \sigma_0} \quad (1)$$

The $T[i][j]$ and $R[i][j]$ represent the $M \times N$ matrix of gray image which deals with grayscale distribution normalization. Template matching is divided into two types: fixed and variable template. The fixed template detects the face depending on the relationship between the correlation of values which are calculated between the pre-established standard template, the detected area and the given rule. Its advantages are that the detection method is intuitive, it requires simple computation and has good robustness, but it cannot adapt to multi-angle, multi-gesture, multi-scale environmental changes. The variable template contains some uncertain factors and parameters and can adjust to the matching template according to the specific conditions that the test sample itself has so as to improve the adaptability and detection accuracy. However, it is difficult to select the deformable template and parameters.

2.2. Face Detection based on SVM

In 1995, Cortes and Vapnik proposed SVM algorithm, which has advantage in dealing with small sample, nonlinear and high dimensional pattern recognition. The SVM algorithm is based on statistical learning theory and structural risk minimization principle, which studies how to construct

the learning machine and how to achieve pattern classification problems, which makes it superior to the other methods in performance. The basic idea is to transform the input space into a high-dimensional space through a nonlinear transformation and to get the optimal linear classification surface in the high-dimensional space in order to solve the non-linear classification problems.

2.3. Face Detection based on AdaBoost Algorithm

Face detection is a process, which determines whether there are face regions in the particular input image or video stream and is required to determine the size, location and posture of the face region if they exist. In 2001, Viola and Paul applied the AdaBoost algorithm to the face detection and proposed the first real-time face detection algorithm. The AdaBoost-based face detection algorithm is an algorithm that is based on integral image, cascade classifier and AdaBoost algorithm.

The basic idea is as following: First it uses the integral image to calculate face's Haar-like features quickly. Then it constructs the weak classifiers that are generated by Haar-like features in accordance with the weighted voting to form a strong classifier using AdaBoost. And lastly it uses the strong classifiers to form a cascade classifier to improve the detection speed of the classification.

2.3.1. Haar-like Features

Haar-like features is divided into edge features, line features and center-surround features (Fig. 1). The features template consists of white and black rectangles. The characteristic value is subtracted between the sum of the pixels filled as white and pixels filled as black. The number of the Haar-like features is based on the matrix size of the training image. Feature template can be placed in any position in the sub-window with any size. Finding all features of sub-window is the basis for training the weak classifier.

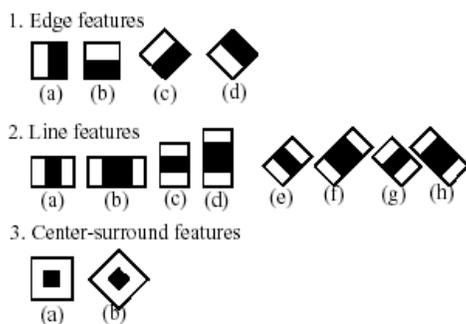


Fig. (1). Haar features image.

2.3.2. Integral Image

After obtaining Haar-like features, how to calculate the rectangular characteristic value becomes a key problem. Because of a large number of training samples, a huge number of the corresponding rectangular characteristic values are generated. The traditional method severely reduces the speed of train and detection. To solve this problem, Viola proposed integral image method to calculate the characteristic values. The integral image is an approach that describes the global information matrix. To calculate the summation of all the

pixels in the image, it only needs to traverse an image once. It is the best way to improve the computational efficiency. The way to construct the integral image is by calculating the summation of all the pixels that are located in the upper left corner of (i, j).

$$ii(i, j) = \sum_{k \leq i, l \leq j} f(k, l) \quad (2)$$

The steps to construct the integral image are as follows:

(1). $s(i, j)$ represents the accumulation of the row direction, the initial $s(i, -1)=0$;

(2). $ii(i, j)$ means an integral image, the initial $ii(-1, i) = 0$;

(3). Scan image by line and recursively calculate the accumulation of each pixel (i, j) by row and the values of integral image $ii(i, j)$.

$$s(i, j) = s(i, j-1) + f(i, j)$$

$$ii(i, j) = ii(i-1, j) + s(i, j)$$

(4). The integral image is constructed successfully when scanning the lower right corner pixel of the image.

2.3.3. AdaBoost Algorithm

AdaBoost is a machine learning approach, which is an iterative algorithm. The core idea is that use the same training set to generate different classifiers (weak classifiers) and combine these weak classifiers into a stronger classifier (strong classifiers). This algorithm is implemented by changing data distribution, each sample weight is confirmed based on whether the classification of each sample in the training set is right or not and whether or not the last overall classification was accurate. The new dataset will be trained by the next classifier. Eventually, all the classifiers will be combined for a last decision classification.

The different training sets of AdaBoost are implemented by adjusting all sample weights. Initially, each sample is given the same weight (The weight shows the probability that the sample is selected into training set by classifier), training a weak classifier is based on this. If the sample is classified correctly, the weight will be reduced, otherwise, the weight will be increased. In this way, the misclassified sample will be highlighted so as to obtain a new sample set. Then the next weak classifier will be generated by training the new sample set which updated the weight. For example it can get a certain number of weak classifiers in an iteration loop. Finally, the strong classifier will be generated by overlapping the weak classifiers according to a certain weight.

The steps of implementing the AdaBoost algorithm are as follows. It sets a given training which S equals to $\{(x_i, y_i) \mid i = 1, 2, \dots, n\}$, $x_i \in X$, $y_i \in Y$, there into X is sample description and Y is sample representation. $y_i \in \{0, 1\}$. In face detection 1 means face and 0 indicates non-face.

(1) It initializes the training sample weights (the initial probability distribution of the training sample). $w_{t,i} = 1/2L$, L is the total of face samples. $w_{t,i} = 1/2M$, M is the total of non-face samples. w_t , t is the error weight of the i th sample in the T time repetition.

(2) For t equal 1 to T (T is the number of weak classifiers).

(3) It normalizes the weights.

$$q_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}} \quad (3)$$

(4) It trains a weak classifier $h(x, f, p, \theta)$ for each feature f and calculate the weighted error rate of the weak classifier.

$$\varepsilon_f = \sum_i q_i |h(x_i, f, p, \theta) - y_i| \quad (4)$$

(5) It chooses the best weak classifier according to the weighted error rate.

$$\varepsilon_t = \min_{f,p,\theta} \sum_i q_i |h(x_i, f, p, \theta) - y_i| = \sum_i q_i |h(x_i, f_t, p_t, \theta_t) - y_i| \quad (5)$$

(6) It updates the sample weight according to the best weak classifier which got by step 5.

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\theta_i} \quad (6)$$

Thus $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$, $\theta_i = 0$ means x_i is classified correctly, $\theta_i = 1$ indicates x_i is classified inaccurately.

(7) It generates a strong classifier by overlapping the weak classifiers on the basis of certain weight.

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{other} \end{cases} \quad (7)$$

2.3.4. Cascade Classifier

In reality, it is often difficult to guarantee the face detection accuracy by only using one strong classifier. Therefore, several strong classifiers will be cascaded in the form of a decision tree to generate a cascade classifier in order to improve the performance of the classifier. In the image detection processing, as long as one classifier is consider that the region of the detection window does not consistent with human facial features, the region will not go to the next judgment. So there are many non-facial areas which will be rejected in the previous processing, while an area that is able to pass all the classifiers will be identified as the face area. This method can improve the classifier accuracy.

2.3.5. The Face Detection Mechanism of AdaBoost Algorithm

The face detection mechanism of the AdaBoost algorithm uses enlarging the detection window approach of multi-scale detection mechanisms rather than reducing the image approach. Because you need to calculate Haar features when the image is reduced each time, which will lead to a large amount of calculation and time consumption. It is difficult to achieve the real-time processing requirements of AdaBoost algorithm. At the beginning, the detection window size is the same as that of the sample size, however the window moves to traverse the entire image area according to a certain scale

parameter and marks the face area. After iterating the entire image, the detection window will be enlarged in terms of specified amplification parameters for the next iteration. The processing will be executed repetitively until the size of detection window is more than the half size of the image.

2.4. AdaBoost Algorithm Result

Using an image as input for the AdaBoost face detection algorithm. Their results are shown in Figs. (2-7). As we can see from the results that AdaBoost has a better performance in detecting faces.



Fig. (2). Face detection results based on Template.

```
Number of Face:1
Face detection run-time:1014ms
```

Fig. (3). Face detection run-time based on Template.



Fig. (4). Face detection results based on SVM.

```
Number of Face:121
Face detection run-time:120058ms
```

Fig. (5). Face detection run-time based on SVM.



Fig. (6). Face detection results based on AdaBoost algorithm.

```
Number of Face:28
Face detection run-time:3370ms
```

Fig. (7). Face detection run-time based on AdaBoost algorithm.

3. ASM ALGORITHM

ASM is an algorithm based on point distribution model. The objects that have similar geometrical shape can be represented by a vector shape which is generated by several key feature points. It can change the shape of the model flexibly to fit the uncertainty characteristics of the target shape. Meanwhile, it can control the shape of the model within the allowable range so as to ensure that the changing of model is not affected by various factors which can lead to the generation of unreasonable shapes. The ASM algorithm mainly includes the following three steps:

3.1. Mark the Key Feature Points and Construct the Shape Vector

Modeling needs to train the key facial areas of face using ASM, N face sample pictures including multi-post and so on will be collected. For each picture, we will manually record the coordinates of several key feature points. Then these K points will form a shape vector:

$$a_i = (x_1^i, y_1^i, x_2^i, y_2^i, \dots, x_k^i, y_k^i), \quad i = 1, 2, \dots, n \quad (8)$$

which (x_j^i, y_j^i) represents coordinate of the j-th key feature point in *i*th sample picture. K means the number of tagged key feature point. N indicates the number of samples. The feature information that is near to the key feature point forms the basis for the match [2].

3.2. Establish Model

Since the face in the picture will be disturbed by external factors such as angle, distance, attitude transform and so on. The shape of human face should be normalized and aligned. The method used here is the procreates method. The basic idea is to align a series of points to the same point distribution without changing the point distribution model through appropriate translation, rotation and scaling transformations. Therefore, it can change the chaotic state of the original data and reduce non-form factor interference.

The ASM algorithm performs the align operations of point distribution model, the steps are as follows:

- (1) It aligns all the face model in the training set to the first face model.
- (2) It calculates the average face model $\bar{\alpha}$
- (3) It aligns all the face models to the average face model $\bar{\alpha}$
- (4) Execute step 2 and step 3 repeatedly until convergence is achieved.

It can be seen that the training sample alignment is an iterative process. After it uses PCA algorithm processes, it aligns the vector shapes. The idea of Principal Components Analysis is doing orthogonal transformation on a sample space, determining the main ingredients and getting main changeable parameters calculated by PCA [2, 3].

- (1) Calculate the average shape vector $\bar{\alpha}$ of all samples shape vector a_i ($i = 1, 2, \dots, n$):

$$\bar{a} = \frac{1}{n} \sum_{i=1}^n a_i \quad (9)$$

- (2) Calculate the covariance matrix of n vectors.

$$S = \frac{1}{n} \sum_{i=1}^n (a_i - \bar{a})^T \cdot (a_i - \bar{a}) \quad (10)$$

- (3) Calculate the characteristic value and the corresponding eigenvectors of the covariance matrix S and sort the characteristic values in descending order to obtain $\lambda_1, \lambda_2, \dots, \lambda_q$ and $\lambda_i > 0$. Choose the previous t eigenvectors from which the corresponding characteristic values should satisfy

$$\frac{\sum_{i=1}^t \lambda_i}{\sum_{s=1}^q \lambda_s} > f_v V_T \quad (11)$$

And f_v is a scale coefficient which is decided by the number of eigenvector and the value is 95%, V_T is the sum of all the characteristic values.

$$V_t = \sum \lambda_i \quad (12)$$

So that any face shape vector used for training can be represented a linear model:

$$a_i \approx \bar{a} + P_s b_s \quad (13)$$

And b_s is a vector including t parameters,

$$b_s(i) = P^T \cdot (a_i - \bar{a}) \quad (14)$$

In order to ensure that the shape caused by b_s is similar with the shape in training set, b_s should be limited by:

$$D_m^2 = \sum_{i=1}^t \frac{b_s(i)^2}{\lambda_i} \leq D_{\max}^2 \quad (15)$$

where $D_{\max} = 3$, if $D_m > D_{\max}$ when b is being updated, b_s should be constrained by the formula(16).

$$b_s(k) = b_s(k) \times \left(\frac{D_{\max}}{D_m} \right) \quad (16)$$

3.3. Construct Local Feature

In order to find new locations for each feature points of every iteration local feature must be built for them. The ASM local feature can be obtained from statistical analysis of the partial window which is with the feature points as the center or the texture sample vectors of the outline in the normal direction. The g_{ij} will be the *i*th feature point of the *j*th sample image. It calculates the mean value and variance of the *i*th feature point;

$$\bar{g}_i = \frac{1}{n} \sum_{j=1}^n g_{ij} \quad (17)$$

$$S_i = \frac{1}{n} \sum_{j=1}^n (g_{ij} - \bar{g}_i)^T \cdot (g_{ij} - \bar{g}_i) \quad (18)$$

Do the same operation on the other feature point to get the corresponding local feature [4].

So the Mahalanobis distance will be used to measure the similarity between the new feature of a feature point and the trained local feature.

$$f_{sim} = (g - \bar{g}_i) S_i^{-1} (g - \bar{g}_i)^T \quad (19)$$

Actually ASM search is an iterative process. The matching between model and the outline of test the image is completed by adjusting parameter b. It changes the position of the model and shapes each iteration in order to detect faces [5].

3.4. ASM Fit

When the ASM model is finished building, one can conduct ASM search operation. First, an initial model is acquired by doing an affine transformation of the average shape, that is to say, the average shape with its center axis progressing counterclockwise θ rotation, zoom s and translate X_c to obtain initial model X .

$$X = M(s, \theta)[a_i] + X_c \quad (20)$$

Then initial model is used to search face in a new image and make the feature points of the search into a final shape that is closest to the corresponding real points.

Actually ASM search is an iterative process. The matching between model and the outline of a test image is completed by adjusting parameter b to change the position of the model and shape in every iteration so as do face detection [6].

4. FACE DETECTION BASED ON ADABOOST AND ASM

Although AdaBoost face detection algorithm has a better performance, because of uncontrollable video and impact of user gestures, the AdaBoost algorithm may regard the non face region as face region easily, which will result in a certain degree of decline in face detection performance and cause a higher rate of face detection. Therefore in the face detection processing, the paper will use ASM algorithm to solve the problems of face detection based on AdaBoost algorithm to improve detection performance [7]. Firstly, the AdaBoost algorithm will be used to detect a face for the very first time, secondly, the detection results will serve as inputs for the ASM algorithm to perform further face region filters to filter non face region which is taken for the face by AdaBoost. Meanwhile, as the symmetry between the two eyes appears as a low gray value prototype, it is relatively easy to extract the eye features. Finally, using ASM model positions the eyes coordinates from face region filtered by ASM [8]. Cutting out the specified size face region on the basis of eyes coordinates paves the way for other steps of face recognition to follow.

4.1. The Steps of Improved Algorithm

The steps of improved algorithm that is used to detect face based on the AdaBoost and ASM algorithm are as follows:

(1) It establishes the ASM model. Firstly, we should manually record the coordinates of 68 feature points (as shown in Fig. 8) for each picture in the sample. Secondly, using the 68 key feature points form a shape vector. Finally, the product method will be used to normalize and align the shape of a human face marked. Use PCA do the data analysis for high dimensional samples to establish the ASM mode (The Fig. 9) shows the examples of ASM model).

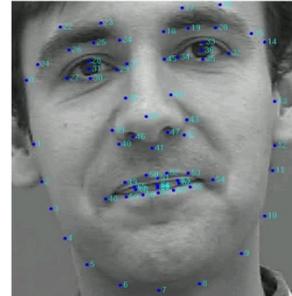


Fig. (8). The 68 key feature points of face.

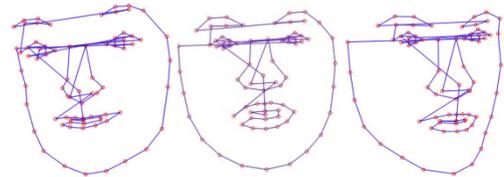


Fig. (9). Examples of the ASM model.

(2) It loads the AdaBoost face detection classifier. Here we introduce the classifier ‘haarcascade_frontalface_alt.xml’ provided by OpenCV, this is a powerful cascade classifier that can improve the face detection rate and detection performance. The loaded path of the cascade classifier ‘haarcascade_frontalface_alt.xml’ can be either the openCV installed path including this file, or the fold path in the project to which the classifier is copied to. Here we chose the second placement to improve the portability of the project.

(3) It inputs the detection images. If the picture is too large, the Gaussian pyramid decomposition method is used to sample and scale down the image. It mainly consists of two steps. First, all the input images are convolved with the specified filters, and then the images are sampled down by denying the even rows and columns to scale the image above a certain size.

(4) It uses AdaBoost algorithm to detect face initially. The system will release the detected image when the face area is still not detected. In the face detection process, the face detection mechanism of the AdaBoost algorithm uses ‘enlarging the detection window’ approach from the multi-scale detection mechanisms rather than using the ‘reducing the image’ approach. At the beginning, the detection window size is the same with the sample size; the window moves to traverse the entire image area according to a certain scaling parameter and marks the face area. After iterating the entire image, the detection window will be enlarged in terms of specified amplification parameters for the next iteration. The processing will be executing repetitively until the size of detection window is more than the half size of the image.

(5) If the 4th step detects face area, then Generate initial shape estimation on the basis of ASM template depending on

the detection frame; Adjust the position of the key characteristic points using the iterative search of ASM, fitting the outline of the face and obtaining the facial feature points (Figs. 10-14);

Firstly, an initial model is obtained by performing the affine transformation of the average shape, that is to say, if the average shape with its central axis doing counter-clockwise rotation, zoom s and translate X_c to obtain initial model X .

$$X = M(s, \theta)[a_i] + X_c \tag{21}$$

Then initial model is used to search face in the picture and make the feature points of the searched final shape are closest to the corresponding real points.

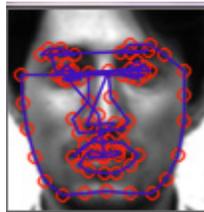


Fig. (10). Example 1 to the outline of face (ASM).

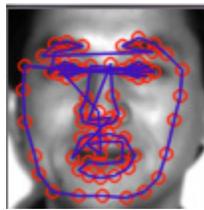


Fig. (11). Example 2 to the outline of face (ASM).

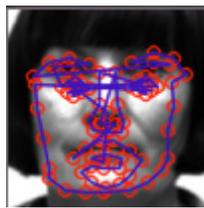


Fig. (12). Example 3 to the outline of face (ASM).

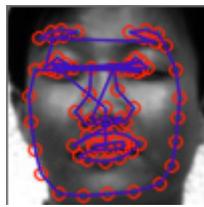


Fig. (13). Example 4 to the outline of face (ASM).

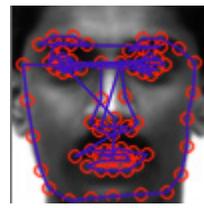


Fig. (14). Example 1 to the outline of face (ASM).

(6) If it is a video steam, it performs the 2nd to 5th steps repetitively until it meets the system requirements; mean-

while, when a face is detected in the video stream, one may hope that the position changes more smoothly. So the eye coordinates will be smoothed between each call.

(7) It acquires the eye coordinates with reference to the position of the feature points of the eyes, the 31 feature point is left eye and the 36 is right eye, the corresponding coordinates are $le(x, y), re(x, y)$;

(8) It calculates the horizontal destinies $xDis$ and the vertical distance $yDis$.

$$xDis = re.x-le.x \quad yDis = re.y-le.y \tag{22}$$

(9) It calculates the distance dis between eyes and the full range of the pitch angle θ using the eyes coordinates and doing the affine transformation of the images based on above results

$$dis = \sqrt{xDis^2 + yDis^2} \tag{23}$$

$$\theta = \arctan\left(\frac{re.y-le.y}{re.x-le.x}\right) \tag{24}$$

(10) Rotate the face image inversely in n degree to make the two eyes are in a horizontal line.

(11) Cut the face rectangular region on the basis of the calculated distant between the two eyes and the facial features and the facial geometric mode (Fig. 15), where the height is $2.2d$, width is $1.8d$, the left and right eye coordinates are $(0.6d, 0.4d), (0.6d, 1.4d)$.

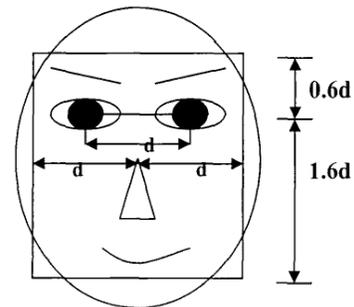


Fig. (15). Face geometric mode.

(12) While the facial proportions are the same cut out according to the 11th steps, the size of each image is deferent from the others. So it is necessary to do face image geometry normalization for the face image cut out to make sure all the images are in the same size. The basic idea of the geometric normalization is to convert the face image in various sizes with respect to the images in a uniform standard size, which contributes to the facial feature extraction. The result is shown in Fig. (16). In this system, the cut out face image is scaled to become the certain size image, 64×84 pixels, by using affine transformation as the input for the next step.

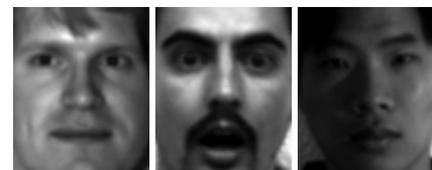


Fig. (16). The result of geometry normalization.

Table 1. Experiment result.

Detection Object	Detection Algorithm	Face Numbers	Accuracy	Detection Rate
Self-built Face database	Adaboost	100	95	95.00%
	Adaboost+ASM	100	98	98.00%
ORL	Adaboost	400	349	87.25%
	Adaboost+ASM	400	362	90.50%
YALE	Adaboost	165	110	66.67%
	Adaboost+ASM	165	125	75.76%

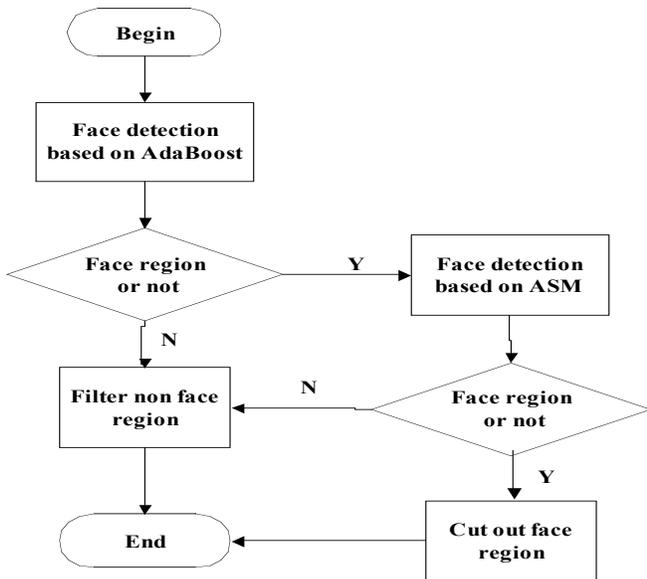


Fig. (17). Flow chart of improved face detection.

The flow chart of improved algorithm face detection shows in Fig. (17):

4.2. Experimental Results and Analysis

This paper uses the initial AdaBoost algorithm and the improved algorithm that uses ASM algorithm on the bases of AdaBoost in the ORL, YALE and self-built face database to detect face for testing. The experimental results are shown in Table 1.

From the table it can be seen that for different test samples, the algorithm that is used in this paper is superior to the original AdaBoost algorithm, which has better robustness in complex environments. Because of the illumination in variation conditions, the facial expressions and facial pose for the YALE face database change more impactful than those of ORL face database. So the detection rate for the YALE face database is less than the rate on the ORL face database. However, overall using the combination AdaBoost and ASM to detect face makes the detection rate has been improved to a certain extent. In summary, in comparison with the traditional AdaBoost face detection algorithm, the results of face detection based on AdaBoost and ASM are significant and stable, and it has good robustness with face illumination, pose and expressional changes. The improved method has a better performance and perfect face detection rate.

5. CONCLUSION

Firstly, the paper studied and analyzed AdaBoost algorithm and ASM algorithm, which showed the AdaBoost algorithm has a better robustness and detection performance compared to other face detection algorithms. However, there will be a certain degree of decline in face detection performance of the AdaBoost algorithm because of unstable videos and from the impact of complex environmental factors. Therefore, it is recommended to use the ASM algorithm to fit the face contour on the basis of AdaBoost algorithm and to cut out the certain sized facial regions. It detects the face in the ORL, YALE and the other face database which is deployed. It draws a conclusion that the improved algorithm has a higher detection rate and robustness.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

The work in this paper has been supported by Beijing Natural Science Foundation (No.4131001). It is also supported by National Natural Science Foundation No. (No.61371143), National Science and Technology Support program (No.2012BAH04F00) and Beijing Natural Science Foundation (No.4132026) and Beijing Education Commission Digital Content Standards and Intelligent Information Processing Platform Project (No.PXM2013_014212_000120) and Beijing Municipal Education Commission Data Acquisition And Visualization Platform Project (No. PXM2014_014212_000017) and 2014 Graduate Innovation Platform Construction Project Funding (No.14085-40).

REFERENCES

- [1] A. Liu, Y. Zhou and X. Guan, "Application of improved active shape model in face positioning", *Computer Engineering*, vol.33, no.18, pp. 227-229, 2007.
- [2] Verma. "Using Viseme Based Acoustic Model for Speech Driven Lip System" In. *IEEE International Conference on Acoustics Speech and Signal Processing* 2003, vol. 5, pp. 720-723.
- [3] Y. Li, C. Zhang, X. Lv., "Face Contour Extraction with Active Shape Models Embedded Knowledge" In. *The 5th International Conference on Signal Processing*. 2000, vol. 2, pp. 1347-1350.
- [4] Y. Fan and J. Ma, "ASM and Improved Algorithm for Facial Feature Location", *Journal of Computer-aided Design & Computer Graphics*, vol.19, no.11, pp. 1411-1415, 2007.
- [5] M. Rezaei, H. Z. Nafchi and S. Morales, "Global haar-like features:

- A new extension of classic haar features for efficient face detection in noisy images". In. *6th Pacific-Rim Symposium on Image and Video Technology*, 2013, pp. 302-313.
- [6] S. Sathyanarayana, R. K. Satzoda, S. Sathyanarayana and S. Thambipillai, "Block-based search space reduction technique for face detection using shoulder and head curves", In. *Image and Video Technology - 6th Pacific-Rim Symposium*, PSIVT 2013, pp. 385-396.
- [7] G. Gibert, D. D'Alessandro and F. Lance, "Face detection method based on photoplethysmography", In: *10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2013, p. 449-453.
- [8] X. Yi, W. Ying and P. Jun, "An improved adaboost face detection algorithm based on the weighting parameters of weak classifier"; In: *Proceedings of the 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing*, 2013, pp. 347-350.

Received: September 22, 2014

Revised: November 30, 2014

Accepted: December 02, 2014

© Bing et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.