Real-Time Detection of Face and Iris

[1]CHAI TONG YUEN, [2]MOHAMED RIZON, [3]MUHAMMAD SHAZRI
[1]Department of Mechatronic & Biomedical Engineering
Universiti Tunku Abdul Rahman (UTAR),
Jalan Genting Kelang, Setapak 53300, Kuala Lumpur,
MALAYSIA
[2]Department of Electrical Engineering, King Saud University,
P.O. Box 800, Riyadh 11421,
KINGDOM OF SAUDI ARABIA
[3]Department of Research and Development
Extol MSC Berhad, Unit G1, Ground Floor,Wisma UOA Pantai,
No. 11, Jalan Pantai Jaya, 59200 Kuala Lumpur,
MALAYSIA
[1]chaity@utar.edu.my, [2]mjuhari@ksu.edu.sa, [3]Shazri.Shahrir@extolcorp.com

Abstract: - In this study, a computational algorithm has been developed to automatically detect human face and irises from color images captured by real-time camera. Haar cascade-based algorithm has been applied for simple and fast face detection. The face image is then converted into grayscale image. Three types of image processing techniques have been tested respectively to study its effect on the performance of iris detection algorithm. Then, iris candidates are extracted from the valley created at the face region. The iris candidates are paired up and the cost of each possible pairing is computed by a combination of mathematical models. Finally, the positions of the detected irises are used as a reference to refine the face region. The algorithm has been tested by quality images from Logitech camera and noisy images from Voxx CCD camera. The proposed algorithm has achieved 83.60% as the highest success rate of iris detection under a user-friendly and unconstraint office environment.

Key-Words: - Face detection, Iris detection, Illumination normalization, Face recognition

1 Introduction

Face recognition has attracted substantial attention from various disciplines and experienced a tremendous growth in researches. Nowadays, face recognition has been widely applied in security system, surveillance, identity verification and other purposes. Automatic face and facial features detections are vital for face recognition development. Humans are good at face identification but it is never an easy task for a computer to recognize human faces automatically.

Automatic real-time face detection systems continue to gain substantial attention in the field of face recognition. Two of such systems have been developed based on still-image [1, 28, 30] and video sequences [29] achieving detection rate of 90% and 99% respectively. These systems used YCbCr or RGB color space in the effort to detect face region. In the still-image system, facial features are searched from face images using thresholding, whereas Radial Basis Function (RBF) neural network is employed to train the video based system to differentiate face and non face regions. However, no significant efforts have been reported in dealing with variabilities such as lighting, facial expressions and head tilts. There is another approach [31] using modular neural networks to have better performance of real-time face detection. This system is able to detect human faces in cluttered scenes. The research has reduced the computational complexity of learning process for large database.

Generally, face recognition is categorized into holistic method and feature-based method. The holistic approach [3, 4] treats a face as a two dimensional pattern of intensity variation. Eyes and mouth are defined as essential facial landmarks which significantly affect the performance of face recognition. The feature-based approach [2, 13, 14] recognizes a face using the geometrical measurements taken among facial features. Many methods have been proposed to locate eyes [8, 32], mouth [4, 5], and face region[3, 7] in an image. Popularity of using template matching and geometrical measurement has been shown at the

earlier stage of feature-based research. R. Brunelli and T. Poggio [2] and D.J. Beymer [3] locate eyes using template matching. However, template matching and eigenspace methods require large number of templates for varying poses and face normalization for variation of size and orientation.

Vertical and horizontal projections are among the most popular methods for geometric-based approach. Projection based methods [2, 6, 8] have been used particularly to find the coarse position of the facial features. This method has been combined by researchers with template matching [2, 3], gabor transformation [22], genetic algorithm [7] and the Smallest Univalue Segment Assimilating Nucleus (SUSAN) [14]. SUSAN algorithm has emerged as a simple and accurate method for eyes, nose and mouth corner detection either in grayscale image [15, 16] or color image [17, 18, 19].Anyhow, integral projection might not robust enough to tackle the illumination changes.

In the most recent researches, a fast and robust facial features detection algorithm has been developed based on discriminative and generative frameworks [23]. Adaboost is used to detect a set of potential features consists of eyes, nose and two lip corners. Facial features are being searched through the entire face region to get all possible combinations of features. The final combination of the features is determined by the maximum likelihood. The detection rate is arguable as a detection of more than three features is considered as a successful detection rather than calculate the success rates separately. Another drawback is the need to zoom the image before running the detection if the features are too small. Besides, a new facial features localization system has been developed based on multi-stream Gaussian Mixture Model (GMM) to locate the regions of eyes, nose and mouth corners [24]. PCA is used to extract each feature from the GMM of each region. One of the weakness of this research is the detection of eyes in the initial stage is not fully automated.

Most of the studies have been done under controlled environment. Conditions such as beard, moustache, spectacle, head orientation, hairstyle, background and facial expressions have been excluded from the face databases used in researches [6, 8, 13, 14, 17, 18, 19]. The aim of this research is to improve the performance of face and iris detection in real-time environment. The proposed algorithm will first locate the face region by using Haar Cascade face detector. Then, illumination normalization is applied on the grayscale image to reduce the effect of lighting variation. After that, iris candidates are extracted from the valleys of the face region using feature template and separability filtering. Iris candidates are determined by the total costs computed from Hough Transform, seperability and intensity information. Finally, the pair of irises will be determined by the total costs and matching correlation in a mathematical approach. Geometric measurements are used to crop the face region again based on the irises positions.

2 Face Localization

There are two databases with images captured by using Voxx CCD box camera (Database A) and Logitech Quickcam Pro9000 (Database B) in office environment. Logitech camera has 2.0 megapixels of resolution while Voxx camera has high resolution 480TVL. Both of the cameras have different build qualities and specifications. From our experiment, the images captured by using Logitech camera have shown better quality, sharpness and less noisy compared to Voxx CCD camera. The Fig. 1 has displayed the example image from both of the cameras respectively.

(a) (b)

Fig. 1: Example images (a) Voxx camera (b) Logitech camera

In this experiment, OpenCV Haar Cascade based face detector [25] has been used for real-time detection instead of the skin-color based face detector as proposed in [30]. This is because the RGB color based face detection tends to be affected easily by the lighting variation and complex background in the office. This is found not that suitable for real-time implementation. The images of each employee are captured during the face verification process before they can open their locker. The images are collected during office hours and no constraint has been set besides facing the camera for verification purpose.

The proposed system is able to choose only one face which is closest to the camera when facing situation with the appearance of multiple faces. The face detection is done under a free environment whereby the source of lighting can be switched on and off during office hours and the users can have

all kind of facial expressions. Eq. (1) is the main function being used in face detection.

$$cvHaarDetectObjects(a,b,c,d,e,f,g) \quad (1)$$

where $a = image$
$b = cascade$
$c = memory\_storage$
$d = scale\_factor$
$e = \min\_neighbours$
$f = flags$
$g = \min\_size$

The parameters d, e, f, g have been set as 1.1, 3, CV_HAAR_FIND_BIGGEST_OBJECT and 65×65 in this experiment. However, there are still unwanted background noises found in the detected face region using this common method. The variation of lighting condition has affected the accuracy of the face detector in eliminating the unwanted noises from background. In this case, an iris detection algorithm has been introduced in Section 4 to refine the detected region to contain only the face.

3 Illumination Normalization

This section describes the chosen image processing technique which is similar to the method proposed by Xiaoyang Tan and Bill Triggs [27] before implementing the iris detection algorithm. It incorporates a series of steps to counter the effects of illumination variations, local shadowing and highlights while preserving the originality of the visual appearance. The first step of this algorithm is gamma correction which is a nonlinear gray-level transformation that replaces gray-level I with I^{γ}, where $\gamma \in [0,1]$ is a user-defined parameter. This can enhance the local dynamic range of the image in dark region while compressing it in bright regions.

The basic principle is that the intensity of light reflected from an object is the product of the incoming illumination and the surface reflectance. In order to recover object-level information independent of illumination, a full log transformation is too strong in practice and tends to over-amplify the noise in dark regions. Anyhow, a power law with the exponent γ in the range of [0, 0.5] is a good compromise to this problem. $\gamma = 0.2$ has been used as the default setting in this experiment. Gamma correction does not remove the influence of overall intensity gradients such as shading effects. It suppresses the highest spatial frequencies and reduces aliasing without destroying too much the underlying signal on which recognition needs to be based.

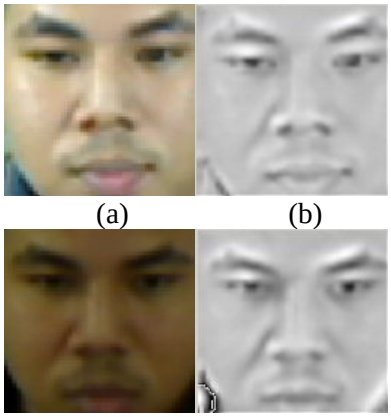
Shading induced by surface structure is potentially a useful visual cue but it is predominantly low frequency spatial information that is hard to separate from effects caused by illumination gradients. The higher spatial frequencies can be suppressed to reduce the aliasing and noise. In practice, it often manages to do so without destroying much of important underlying signal for recognition purpose. Difference of Gaussian (DoG) Filtering is a convenient way to obtain the resulting bandpass behavior. The inner Gaussian is typically set to quite narrow $\sigma_0 \le 1$ pixel for fine spatial detail while the outer one might have σ_1 of 2-4 pixels, depending on the spatial frequency at which low frequency information becomes misleading rather than informative. $\sigma_0 = 1.0$ and $\sigma_2 = 2.0$ are used as default setting.

Sometimes, a mask is needed to suppress the facial regions which contain too much irrelevant information. Contrast Equalization is the final step to rescale the image intensities and measure the overall contrast or intensity variation. A robust estimator needs to be used because the signal typically still contains a small admixture of extreme values produced by highlights at the image borders and small dark regions such a nostrils. Two stages of processes as shown in Eq. (2) have been proposed:

$$I(x,y) \leftarrow \frac{I(x,y)}{\left(mean\left(|I(x',y')|^{a}\right)\right)^{1/a}}$$
$$I(x,y) \leftarrow \frac{I(x,y)}{\left(mean\left(\min\left(\tau,|I(x',y')|\right)^{a}\right)\right)^{1/a}} \quad (2)$$

Here, α is a strongly compressive exponent that reduces influence of large values, τ is a threshold used to truncate large values after the first phase of normalization and the mean is over the unmasked part of the image. $\alpha = 0.1$ and $\tau = 10$ are used in this experiment. Finally, a nonlinear function is being used to compress over-large values although the resulting image is well scaled in previous step. This is to reduce the influence of those extreme values on subsequent stages of processing. The hyperbolic tangent has been used $I(x,y) \leftarrow \tau \tanh(I(x,y)/\tau)$, thus limiting I to the range $(-\tau, \tau)$.

This algorithm has been tested on its robustness under various lighting conditions. From Fig. 2, the normalized images have shown similarity in terms of illumination distribution even though the input images are taken under different lighting conditions.



(a) (b)

(c) (d)

Fig. 2: (a) and (c) are input images; (b) and (d) are the respective output images after illumination normalization

Besides Histogram equalization, LogAbout has been compared with the proposed normalization method to evaluate the performance of iris detection based on the processed images. This technique has been proven to have better performance than histogram equalization in [26]. Nonlinear transformations (Log) are often used in computer vision. There is physiological evidence that the cell's response in the retina is nonlinear in the intensity of the incoming image, which can be approximated as a Log function of the intensity. The form of the Log transformation is shown as below:

$$g(x,y) = a + \frac{\ln\left(f(x,y)+1\right)}{b\ln c} \tag{3}$$

Where $f(x,y)$ is an original image, while a,b and c are parameters which can control the location and shape of the logarithmic curve.

According to the research, Log function is able to improve the illumination deficient in essence. Log transformation is useful for shadow and non-uniform image. LogAbout is then introduced because Log transform does compress high gray pixels causing possible lost of some edge information. In LogAbout, a high pass spatial filter as shown in Table 1 has added before Log transform to evaluate the brightness of group of pixels. High pass filter can remove low-frequency components while retaining the high-frequency features of interest, thus reduce the effects of non-uniformities. This will result a sharper image in return after LogAbout processing.

Table 1: High pass filtering mask

-1	-1	-1
-1	9	-1
-1	-1	-1

4 Iris Detection Algorithm

The proposed algorithm will first apply grayscale conversion on the input image followed by grayscale closing [12]. Then, illumination normalization and light spot deletion are applied to remove the reflection of lighting on the irises. This can be done by replacing the center pixel of a mask by the smallest intensity value of the pixels within the mask. Histogram equalization, LogAbout [26] and illumination normalization have been tested and compared in this experiment to reduce the effect of lighting variation.

4.1 Valley extraction

After the illumination normalization, valley extraction as shown in Eq. (4) has been carried out. Each pixel (x,y) in the face region:

$$V(x,y) = G(x,y) - I(x,y) \tag{4}$$

where $G(x,y)$ and $I(x,y)$ denote the value obtained from grayscale closing and intensity value. Region which consists of pixels (x,y) such that $V(x,y)$ is greater than or equal to a threshold value are determined to be valleys.

$$\frac{1}{N}\sum_{i=0}^{MAX} h(i) \geq p \tag{5}$$

N is the number of pixels in the face region. MAX is the maximum of $V(x,y)$ over all pixels (x,y) in the face region. $h(i)$ is the number of

pixels in the face region within $0 \le i \le MAX$. p is a parameter given as input. (p is set to 0.1 for all the images used in this experiment)

4.2 Selection of iris candidates

In irises selection, this algorithm performs similar method as proposed in [30]. First, it computes the costs $C(x,y)$ for all pixels in the valleys and selects m pixels according to non-increasing order that give the local maxima of $C(x,y)$ as the iris candidate locations.

$$C(x,y) = C_1(x,y) + C_2(x,y) \tag{6}$$

Where $C_1(x,y)$ is the mean crossing of the row and column pixels and $C_2(x,y)$ is the intensity difference between the center part and the boundary part of a square region. An eye template as shown in Fig. 3 is placed at each candidate location and measures the separability [10] between the two regions R_1 and R_2 given by:

$$\eta = B/A \tag{7}$$

Where $A = \sum_{i=1}^{N} (I(x_i, y_i) - \overline{P}_m)^2$,

$B = n_1(\overline{P}_1 - \overline{P}_m)^2 + n_2(\overline{P}_2 - \overline{P}_m)^2$,

$n_k(k=1,2)$: number of pixels in R_k ,

$N = n_1 + n_2$,

$\overline{P_k}$ (k = 1,2): average intensity in R_k ,

$\overline{P_m}$: average intensity in the union of R_1 and R_2 ,

$I(x_i, y_i)$: the intensity values of pixels (x_i, y_i) in the union of R_1 and R_2 .

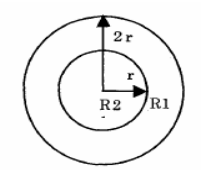


Fig. 3: An eye template to detect blob

Next, it applies Canny edge detector [11] to the face region and measures the fitness of those iris candidates to the edge image using Hough Transform [21]. The vote calculated from Hough Transform is defined as $C_1(i)$. Given an iris candidate $B_i = (x_i, y_i, r_i)$, measures the fitness of iris candidates to the intensity image by placing two templates as shown in Fig. 4. Compute the separabilities $\eta_{23}(i)$, $\eta_{24}(i)$, $\eta_{25}(i)$ and $\eta_{26}(i)$ using Eq. (5) for $C_2(i)$ and $C_3(i)$. $C_4(i)$ is the ratio of average intensity of each iris candidate over average intensity of all the candidates.

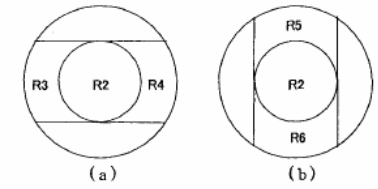


Fig. 4. The templates used to compute $C_2(i)$ and $C_3(i)$

Then the cost for each iris candidate is calculated as below:

$$C(i) = C_1(i) + C_2(i) + C_3(i) + C_4(i) \tag{8}$$

Where $\mathrm{C_1(i)} = \dfrac{\mathrm{V_{max}}}{\mathrm{V(i)}}$

$$\mathrm{C_2(i)} = \frac{|\,\eta_{23}(\mathrm{i}) - \eta_{24}(\mathrm{i})\,|}{\eta_{23}(\mathrm{i}) + \eta_{24}(\mathrm{i})}$$

$$\mathrm{C_3(i)} = \frac{|\,\eta_{25}(\mathrm{i}) - \eta_{26}(\mathrm{i})\,|}{\eta_{25}(\mathrm{i}) + \eta_{26}(\mathrm{i})}$$

$$\mathrm{C_4(i)} = \frac{\mathrm{U(i)}}{\mathrm{U_{av}}}$$

in which $V(i)$ is the vote for $\mathrm{B_i}$ given by Hough transform; $V_{\max}$, the maximum of $V(i)$ over all iris candidates; $U(i)$ is the average intensity inside B_i and U_{av} is the average of $U(i)$ over all iris candidates.

Finally, computes the cost for each pair of iris candidates B_i and B_j :

$$F(i,j) = t\{C(i) + C(j)\} + (1-t)/R(i,j) \tag{9}$$

Where $C(i)$ and $C(j)$ are costs computed by Eq. (8). $R(i, j)$ is the normalized cross-correlation value computed between eye template and the input image. The eye template used in this experiment is shown in Fig. 5 t is the weight to adjust two terms of the cost. The lowest cost $F(i, j)$ is selected as the pair of irises as shown in Fig. 5.

Fig. 5: The eye template used for computation of $R(i, j)$

5 Iris based face cropper

The coordinates of the selected irises are used in this algorithm to refine the detected face region. In order to eliminate background regions and contain only face, some geometric calculations have been proposed. Let x_1, x_2, y_1, y_2 denote the x-coordinates and y-coordinates of the left and right irises.

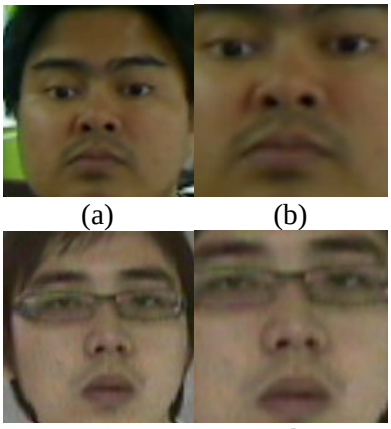
$$x_0 = (x_2 - x_1)/2$$
$$y_0 = (y_1 + y_2)/2 \quad (10)$$

Where x_0 and y_0 are the horizontal and vertical ratios which will be used for geometric calculations. After obtaining the positions of irises and ratio values, a boundary which contains only face can be estimated based on the geometric equations as below:

$$xl = x_1 - x_0 + \Delta x$$
$$xr = x_2 + x_0 - \Delta x$$
$$width = xr - xl$$
$$y\min = y_0 - x_0 - \Delta y$$
$$y\max = y\min + width + \Delta y \quad (11)$$

Where $xl, xr, y\min, y\max$ are the left, right, upper and lower coordinates of the face boundary. Δx and Δy are two variables which can be set based on different kind of face databases ($\Delta x = 5, \Delta y = 6$ are used in this experiment). Three rules have been defined to detect the failure cases of iris detection. The Euclidean Distance between the detected irises, the center y-position of the face image and the orientation angles of the detected irises. An iris detection will be classified as failure if the one of the parameters as mentioned has been exceeded. As a result, the system will use the face region detected by haar cascade face detector. Otherwise, the detected face region will be refined based on the irises positions.

From Fig. 6, there are example images from both of the cameras. Fig. 4(a) and (c) show the outputs of the detected face region by using haar cascade based face detector. There are some noticeable background noises in the images. However, these can be eliminated based on the geometric calculations as proposed in this research. The results of the iris-based face cropping are shown in Fig. 4(b) and (d).



(a) (b)

(c) (d)

Fig. 6: (a) and (c) are outputs of haar cascade face detector; (b) and (d) are outputs from iris based face cropper

6 Result and discussion

Experiments have been done to evaluate the real time performance of the proposed iris detection algorithm with histogram equalization, LogAbout and illumination Normalization. A total of 109 and 122 images captured by Voxx and Logitech cameras are collected for this experiment under office environment. One image from Voxx camera and three images from Logitech camera have been eliminated from the databases since one of the iris is not appeared in the images. This research intends to make the system user-friendly whereby users have not been limited in wearing spectacles, type of clothes, hairstyles, facial expressions, multiple faces and background. Here, Database A represents the images captured by Voxx camera while Database B represents the images captured by Logitech camera.

From Table 2, histogram equalization, LogAbout and illumination normalization scored

38.53%, 54.13% and 68.81% respectively for database A for success detection. The success rates of iris detection for database B are 62.30% (histogram equalization), 71.31% (LogAbout) and 83.60% (illumination normalization). Thus, Illumination normalization has achieved the highest iris detection rate in both databases at real-time followed by LogAbout and histogram equalization. This means the images processed by illumination normalization have better contrast and stability when dealing with lighting variation. However, all of the processing techniques do not show outstanding results when dealing with database A which are noisier and blur images.

Table 2: Iris detection rates for different image processing techniques for Database A and Database B

Processing Technique	Database A (%)	Database B (%)
Histogram Equalization	38.53	62.30
LogAbout	54.13	71.31
Illumination Normalization	68.81	83.60

Table 3: Improvement reported on iris detection rate and background noise reduction

Processing Technique	Improvement (%)		Background Noise (%)	
	A	B	A	B
LogAbout	15.60	9.02	7.99	8.04
Illumination Normalization	30.28	21.30	1.32	1.31

From Table 3, the proposed processing technique has improved the performance of iris detection more than 30% (Database A) and 21% (Database B) compared to the histogram equalization which was proposed in previous research. LogAbout has reported an improvement of 15.60% and 9.02% for Database A and Database B. Although the improvement rate for Database A is great but the overall success rate is still considered low for all the processing methods with the highest detection rate of 68.81%. Anyhow, illumination normalization outperforms LogAbout to nearly 15% when processing the low quality images from Database A.

An average of 9.30% pixels of unwanted background noise are reported in the detected face region of Voxx and Logitech images using Histogram equalization as the image processing method. Since the final cropping of the face region is based on the detected irises position, optimization in iris detection performance has indirectly reduced the noise pixels from the background. There are only averagely 1.32% and 1.31% of background noise found from images in Database A and Database B after the implementation of illumination normalization. The integration of illumination normalization has made the iris detection algorithm more efficient with 8% reduction of unwanted background pixels compared to histogram equalization. Not much background pixels have been reduced by LogAbout compared to histogram equalization since the improvements in both databases are not that significant.

Successful iris detections from both databases are shown in Fig. 7. This has covered a variety of lighting conditions, genders, backgrounds, spectacles, expressions and hairstyles. The improved iris detection algorithm has shown its robustness through this experiment especially when facing issues such as complex backgrounds and multiple faces. However, there are also some failures reported. The examples of the fail detection are shown in Fig. 8. Most of the subjects in those fail detections are wearing spectacles. The proposed algorithm has not been successful when dealing with certain types of spectacles which have some levels of lens reflectance when expose to lighting. This explains why there are also successful iris detections for the same subjects under different lighting conditions.

By studying the fail cases, frame of a spectacle can cause the iris detection algorithm to fail in some situations. This tends to happen when the subject wears the spectacle slightly lower than its normal position. The upper frame will block or appear at the same line as the irises horizontally. This has confused the proposed algorithm in making the correct decision especially when the color of the frame is black which has similar intensity values with the iris. Besides, spectacle with low transparency affects the visibility of iris. This becomes worse in darker lighting condition, low quality image and the subject has some distances from the camera. Another possibility that can cause fail detection is the appearance of irises. Some facial expressions of the subjects have caused the eyes to be nearly closed and indirectly causing the irises not

visible enough for detection. One of the failure case in Fig. 8 shows that the thick moustache of a subject under such lighting condition might confuse the detection algorithm. Solutions in terms of mathematical model or rules need to be set to differentiate the features which have similar intensity value as irises.

The integration of illumination normalization into the iris detection algorithm replacing histogram equalization has improved the performance of the proposed algorithm in real-time. Besides, the ability of this algorithm to deal with the appearance of spectacles has been increased.

Fig. 7: Successful iris detections

Fig. 8: Fail iris detections

5 Conclusion

The experimental result has proven that the performance of the iris detection algorithm can be improved more tan 30% by applying a better image processing technique. Illumination normalization has proven its ability to cope with lighting variation under office environment. The proposed iris detection algorithm has achieved a success rate of 83.60% in a user-friendly real-time environment. Through this experiment, the algorithm has shown its robustness when tested with subjects wearing different kind of spectacles. Besides, the irises positions have been found useful to re-justify the boundary of face region detected by using OpenCV face detector. The iris based face cropper as proposed has successfully excluded the unwanted background pixels to only 1.31%.

In future, the iris detection algorithm needs to be enhanced in order to perform better under conditions like lens reflectance, facial expressions, disruption of moustache and excessive lighting. Finally, official online databases will be used to benchmark and evaluate the performance of the proposed algorithm in near future.

References:

[1] Tsuyoshi Kawaguchi and Mohamed Rizon, Iris detection using intensity and edge information, *Pattern Recognition 36*, 2003, pp. 549-562.

[2] R Brunelli and T. Poggio, Face recognition: features versus templates, IEEE *Trans. Pattern Anal. Mach. Intell. 15 (10)*, 1993, pp. 1042-1052.

[3] D. J. Beymer, Face recognition under varying pose, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, 1994, pp.756-761.

[4] Chun-Ming Li, Yu-Shan li, Qing-De Zhuang and Zhong-Zhe Xiao, The face localization and regional features extraction, *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, 2004, pp. 26-29.

[5] Karin Sobottka and Ioannis Pitas, A novel method for automatic face segmentation, facial feature extraction and tracking, *Signal Processing: Image Communication 12*, 1998, pp. 263-281.

[6] Selin Baskan, M. Mete Bulut and Volkan Atalay, Projection based method for segmentation of human face and its evaluation, *Pattern Recognition Letters 23*, 2002, pp. 1623-1629.

[7] Kwok-Wai Wong, Kin-Man Lam and Wan-Chi Siu, An efficient algorithm for human face detection and facial feature extraction under different conditions, *Pattern Recognition 34*, 2001.

[8] Yeon-Sik Ryu and Se-Young Oh, Automatic extraction of eye and mouth fields from a face image using eigenfeatures and multilayer perceptrons, *Pattern Recognition 34*, 2001, pp. 2459-2466.

[9] C.H. Lin and J. L. Wu, Automatic facial feature extraction by genetic algorithms, *IEEE Trans. Image Process. 8 (6)*, 1999, pp. 834-845.

[10] K. Fukui and O. Yamaguchi, Facial feature point extraction method based on combination of shape extraction and pattern matching, Trans. *IEICE Japan J80-D-II (8)*, 1997, pp. 2170-2177.

[11] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell. 8 (6)*, 1986, pp. 679-698.

[12] S. R. Sternberg, Grayscale morphology, *Computer Vision Graphics Image Process. 35*, 1986, pp. 333-355.

[13] Hua Gu, Guangda Su and Cheng Du, Feature points extraction from faces, *Image and Vision Computing NZ*, Palmerston North, 2003.

[14] Yu Song, Kun He, Jiliu Zhou, Zhiming Liu and Kui Li, Multi-resolution feature extraction in human face, *Proceedings of International Conference on Information Acquisition*, 2004.

[15] Stephen M. Smith and J. Michael Brady, SUSAN – A new approach to low level image processin, *International Journal of Computer Vision 23(1)*, 1997, pp.45-78.

[16] Mauricio Hess and Geovanni Martinez, Facial feature extraction based on the Smallest Univalue Segment Assimilating Nucleus (SUSAN) algorithm, *Image Processing and Computer Vision Research Lab (IPCV-LAB)*, Escuela de Ingenieria Electrica, Universidad de Costa Rica.

[17] Yulu Qi, Goenawan Brotosaputro and Nuanwan Soonthomphisaj, Finding the estimated position of facial features on the human face using intensity computation, *IEEE Asia-Pacific Conference (APCCAS)*, 1998, pp. 579-582.

[18] Vladimir Vezhnevets, Face and facial feature tracking for natural human-computer interface, *International Conference Graphicon*, Nizhny Novgorod, Russia, 2002.

[19] Jie Yang and Alex Waibel, A real-time face tracker, *Application of Computer Vision, WACV'96*, 1996, pp. 142-147.

[20] A. M. Martinez and R.Benaventa, The AR face database, *CVC Technical Report*, No.24, June 1998.

[21] Tsuyoshi Kawaguchi, Daisuke Hidaka, Mohamed Rizon, Detection of eyes from human faces by hough transform and separability filter, *Department of Computer Science and Intelligence Systems*, Oita University, Oita, Japan, pp. 870-1192.

[22] Y. Zhao, X. Shen and N.D. Georganas, Combining integral projection and gabor transformation for automatic facial feature detection and extraction, *Proceedings of IEEE International Workshop on Haptic Audio Visual Environment and their Applications*, 2008, pp. 103-107.

[23] V. Erukhimov and K.C. Lee, A bottom-up framework for robust facial feature detection,

Proceedings of 8th International on Automatic Face and Gesture Recognition, 2008, pp.1-5.

[24] K. Kumatani, H.K. Ekenel, H. Gao, S. Rainer, E. Aytul, Multi-stream gaussian mixture model based facial feature localization, *Proceedings of Signal Processing, Communications and Applications Conference*, 2008, pp. 1-4.

[25] P. Viola and M.J. Jones, Robust real-time face detection, *International Journal of Computer Vision 57(2)*, 2004, pp. 137-154.

[26] H. Liu, W. Gao, J. Miao and J. Li, A novel method to compensate variety of illumination in face detection, *Joint Conference on Information Sciences*, 2002, pp. 692-695.

[27] Xiaoyang Tan and Bill Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, *IEEE International Workshop on Analysis and Modelling of Faces and Gestures*, 2007, pp. 235-249.

[28] W.J. Chew, L.M. Ang and K.P. Seng, Automatic model based face feature detection system, *Proceedings of International Symposium on Information Technology*, 2008, pp. 1-6.

[29] M. Balasubramaniam, S. Palanivel, and V. Ramalingam, Real time face and mouth Recognition using radial basis function neural networks, *Expert Systems with Applications* 2008, pp. 1-8.

[30] Chai Tong Yuen, M. Rizon, Woo San San and M. Sugisaka, Automated detection of face and facial features, *The 7th WSEAS Int. Conference on Signal Processing, Robotics and Automation*, University of Cambridge, United Kingdom, 2008, pp. 230-234.

[31] Hazem M. El-Bakry and Nikos Mastorakis, A new approach for fast face detection, *Proceedings of 7th WSEAS Int. Conference on Neural Networks*, Cavtat, Croatia, 2006, pp. 152-157.

[32] Ching-Tang Hsieh, Eugene Lai, Chi-Liang Shen and Yeh-Kuang Wu, A simple and effective real-time eyes detection human detection without training procedure, *Proceedings of The 6th WSEAS Int. Conference on Signal, Speech and Image Processing*, Libson, Portugal, 2006, pp. 215-220.