A PC Camera Based Method for Extracting Facial Features And Its Application To Distraction Detection In E-Learning

A PC Camera Based Method for Extracting Facial Features And Its Application To Distraction Detection In E-Learning

Fan Xiaomeng
Graduate School of Computer and Information Sciences
Hosei University
Tokyo 184-8584, Japan
xiaomeng.fan.39@stu.hosei.ac.jp

***Abstract*—As the e-learning is becoming more and more popular and widely used all over the world, many schools and companies start to use e-learning to teach lessons online. While, in despite of its convenience and high speed, managers soon realize that it is very important to find a reliable and feasible method of judging the performance of the learner to get feedback with which managers can monitor behaviors of learners just like a teacher can do in ordinary classes. Existing method to evaluate performance of the learner can be roughly divided into two groups: evaluation of result based and evaluation of behavior based. However, most of these methods fail to reflect the true study status of the learner. In this paper, an image processing technique based method to estimate the head pose of the learner with which the system will judge the distraction on the learner is proposed. It uses a single PC camera to capture the facial information of the learner and then gets positions of facial features by analyzing the facial image. Then use position information to estimate the roll angle of the head and the horizontal yaw angle of the face in order to calculate the key index to judge the distraction rate of the learner with a designated formula. A series of experiments show that despite the accuracy of head pose estimation is not so well (about 40%), the success rate of judging distraction status can reach nearly 80% if the head pose estimation succeed.**

Keywords: Face Recognition, Head Pose Estimation, Skin Detection

I. INTRODUCTION

The e-learning, the concept of which is a computer based educational tool or system that enables user to learn anywhere through the internet, has played an important role in education and it is becoming more and more popular and acceptable with high-speed development and widespread of the network construction. However, despite of having a series of advantages such as flexibility and globality, one of the biggest disadvantage is that it is hard to judge the study performance of learners precisely because there is no face-to-face teacher who will monitor them like traditional class [1]. As a result, different kinds of methods are developed to solve this problem. Someone choose to create an isolated monitoring system based on an evaluation model and generate performance report [2]. And some researchers try to connect a monitoring system to the e-learning system to complete the procedure [3]. While these mentioned approaches have a bad performance on real-time feedback.

In this paper, an automatic monitoring method based on image processing technique is proposed. It uses the ordinary PC camera which is already widely used on PC to gain the image of the learner sitting in front of the screen. The image will be firstly filtered by implementing the skin detection rule introduced by Peer et al. [4], after which the image's specified areas matching skin detection rules can remain their status while other areas will be converted to gray color. Then the erosion and dilation operations are adopted to smooth the image and reduce noises. Then a function will scan the image and erase those objects with inappropriate size (too big or too small). The face area will be then extracted as an isolated image for processing, with which the system starts to locate facial features including eyes, mouth and nose. The face area will be firstly transferred into binary format, under which the skin area is replaced with white pixel and facial features is replaced with black, and then divided into two parts: eye and mouth parts. The system will traverse the eye area and try to locate two objects in black, if they just match some specified rules, they will be recognized as eyes. After locating eyes positions, positions of pupil can be obtained by implementing the method to calculating the gradient value around eye. The method to locate the mouth uses similar theory as eye: try to find the longest object in the field. The area for locating the nose will be extracted based on locations of eyes and mouth, which means a rectangle between two eyes and mouth is extracted, where the nose tip position is obtained by using gradient based method. Then the roll angle of the head and the face offset index will be calculated with positions of facial features. With these two parameters, the distraction index can be calculated by using a formula which will be introduced in part IV. Finally, the status of learner will be judged with this distraction index.

II. RELATED WORK

A. Introduction of Current Approaches

In order to detect distraction with angles of face and head, a very popular and interesting research area is imported: the head pose estimation. Technically, current approaches for the head pose estimation can be divided into two groups: 2D-based and

3D based. Because the source data of our research based on 2D images, only 2D-based approaches will be introduced.

The 2D-based approaches can be roughly categorized into two main types: facial features-based and facial appearance-based approach.

Generally speaking, facial features-based methods firstly try to locate facial features on face area, such as eyes, nose and mouth, then use these position information to estimate head pose. To detect facial features, some researchers use the classifier with AdaBoost Algorithm like [5, 11, 12]. According to these papers, we can see that researchers may usually need to train multiple classifiers not only for different facial features but also for variations in angle and illumination of them in order to achieve an acceptable result. The accuracy of the experiment generally depends on the quality and quantity of classifiers, which means researchers will spend lot of time on collecting sample images for classifiers and training them. In the paper [13], a skin color model based method is proposed. They firstly obtains the map of facial features based on YCbCr skin color model. Then, they does the binary operation of the map and search the four connected regions based on binary result to construct the candidate facial features regions.

While appearance-based methods mainly use the global facial features such as the whole shape of the face. Some approaches use multiple cascades that are trained for different head poses [6], which means there will be a group of cascades and each cascade is responsible for a designated angle range of head pose. The magnitude of the angle range determines the number of cascades and the final accuracy of the algorithm. Someone choose to train a tree of classifiers using hierarchically sub-sampling the pose space [7], which means the classifiers work by levels. The appearance-based approach is widely used in processing those low resolution images.

III. FACIAL FEATURES DETECTION

In this chapter we will introduce and explain methods and algorithms used in our approach detailedly. The rule for skin detection will be firstly introduced in section A. The section B explains how to get the face area obtained by skin detection rule. In section C we will talk about the method of getting positions of eyes and mouth and the way of getting position of nose is going to be discussed In section D.

A. Skin Detection Rule

After receiving the input image, the first step is to implement the skin detection rule on the image. According to the research done by Garcia and Tziritas in their paper [8], the process of skin detection should be implemented in all three color spaces: RGB, YCbCr and HSV.

In RGB space, the rule introduced by Peer [4] is used. The skin color under normal daylight illumination is described as:

$$(R > 95)\ \text{AND}\ (G > 40)\ \text{AND}\ (B > 20)\ \text{AND}\ (\max\{R, G, B\} - \min\{R, G, B\} > 15)\ \text{AND}\ (|R - G| > 15)\ \text{AND}\ (R > G)\ \text{AND}\ (R > B) \quad (1)$$

And there is another rule for skin color under lateral light illumination:

$$(R > 220)\ \text{AND}\ (G > 210)\ \text{AND}\ (B > 170)\ \text{AND}\ (|R - G| \leq 15)\ \text{AND}\ (R > B)\ \text{AND}\ (G > B) \quad (2)$$

In reality, the image may match one of these two rules, so the logical relationship in RGB space is (1) OR (2).

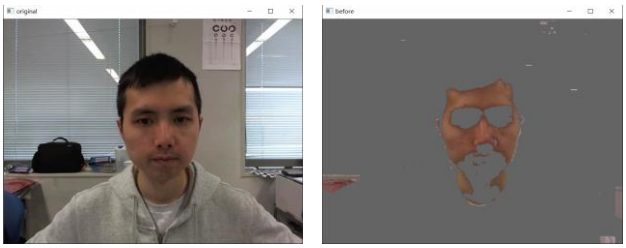
In YCbCr space, there are 5 bounding rules according to this paper [9], and the skin color must match them all:

$$Cr \leq 1.5862 \times Cb + 20\ \text{AND}\ Cr \geq 0.3448 \times Cb + 76.2069\ \text{AND}\ Cr \geq -4.5652 \times Cb + 234.5652\ \text{AND}\ Cr \leq -1.15 \times Cb + 301.75\ \text{AND}\ Cr \leq -2.2857 \times Cb + 432.85 \quad (3)$$

In HSV space, the skin color should match one of these two rules:

$$H < 25\ \text{OR}\ H > 230 \quad (4)$$

After implementing the skin detection rule, we can get the rough skin area in the image (Left side is input image):

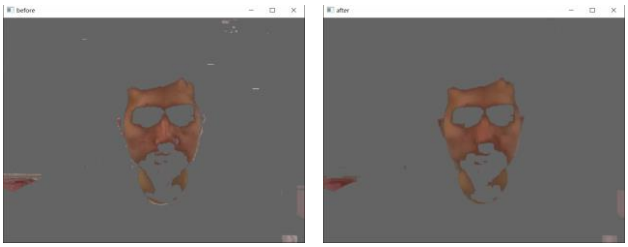


(a) (b)

Fig. 1. Source image from camera (a), skin detection result (b).

B. Find the Face in Image

After we get the skin area of the image, the first step is to use operation of the erosion and dilation. Erosion is one of two fundamental operations in morphological image processing from which all other morphological operations are based. It was originally defined for binary images, later being extended to grayscale images, and subsequently to complete lattices. While Dilation is another basic operations in mathematical morphology. The dilation operation usually uses a structuring element for probing and expanding the shapes contained in the input image. After implementing these two operations, we can get another image with little noises (Left side is original image):

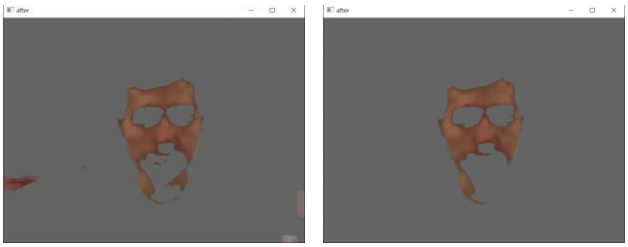


(a) (b)

Fig. 2. Image before filter (a), result after erosion and dilation (b).

Then the system will scan the image and filter those unnecessary parts with a threshold value. Because we assume that there is only one learner sitting in front of the screen. The logic of the operation is those parts whose pixel sizes are below this value are going to be filtered. According to tests in the lab,

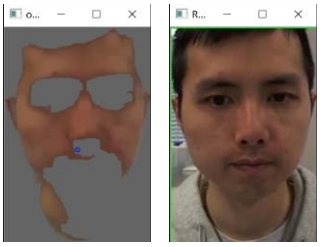
this value should be set to 6000. After implementing this operations, we can get another image with almost just one big skin area (Left side is original image):



(a) (b)

Fig. 3. Source image (a), result after filter (b).

Now the system can get the biggest skin object in the image and treat it as the face area. A rectangle area will be created based on the face area's top node on the left and bottom node on the right. Then a new image within this rectangle is extracted from the original RGB image for further process:



(a) (b)

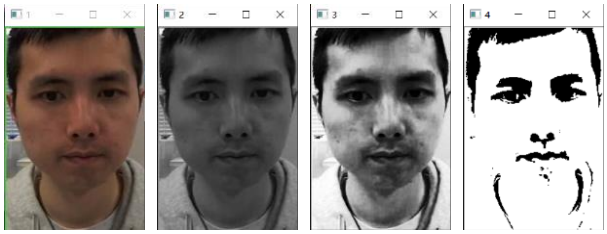
Fig. 4. Extracted skin area (a), extracted face area (b).

C. Detect Eyes and Mouth

The obtained face image will be firstly transferred to grayscale format. Then the histogram equalization is carried out on the image to make contours of facial features more clear. Then the equalized image is processed by the binary thresholding operation which means all pixels' gray value in the image will be converted to 0 if their gray value is lower than a given threshold value or converted to 255.

$$\mathrm{dst}(x,y) = \begin{cases} maxVal, & \text{if } \mathrm{src}(x,y) > thresh \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

In here, the dst(x, y) is a pixel value. The maxVal in grayscale format image means 255. By this, there will be only the black and white color in the image:



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

Fig. 5. Source face image (a), grayscale format (b), grayscale format after histogram equalization (c), and binary format (d).

In the next step, we will use the algorithm provided by J Shi and C Tomasi [10] to extract all corner points using the grayscale image as the input. Here is some simple explaination of the algorithm: we can assume a grayscale 2-dimensional image is used. Let this image be given by *I*. Consider taking an image patch over the area *(u,v)* and shifting it by *(x,y)*. The weighted sum of squared differences (SSD) between these two patches, denoted *S*, is given by:

$$S(x,y) \approx (x\ y)A\begin{pmatrix} x \\ y \end{pmatrix} \quad (6)$$

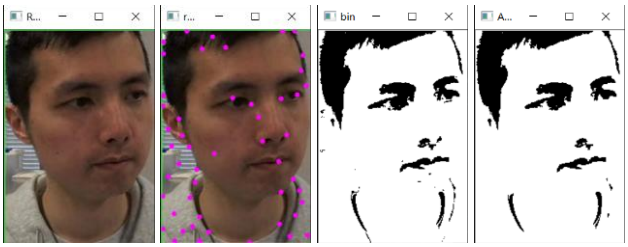
where A is the structure tensor:

$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \quad (7)$$

A should have two "large" eigenvalues for an interest point. Based on the magnitudes of the eigenvalues, a judgement value will be get from $min(\lambda_1, \lambda_2)$. If this value is greater than a threshold value, it will be considered as a corner point.

We can use the function goodFeaturesToTrack in OpenCV lib to use this algorithm to get corner points in the image.

As we can see in the Fig.6, almost all facial features are marked by one or more corner points. These corner points will be stored and later used for the refinement of the binary image. The logic of refinement is simple: all areas consists of black pixels in the binary image area are viewed as objects. If an object has at least one corner point inside or close to it, it will be kept or it will get erased.



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

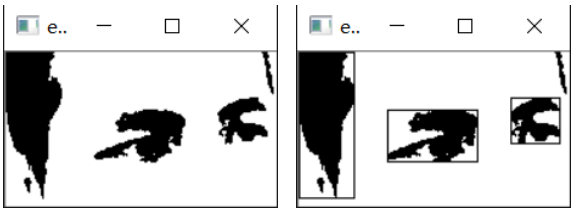
Fig. 6. Source image (a), image with corner points (b), binary image (c), and filter result (d).

It is known to all that eyes locate in the upper part of the face and the mouth in the lower part. According to this fact, the binary image will be divided into two parts, eye part and mouth part, by two ranges of height. The eye part's height is from height * 0.102 to height * 0.488 and the mouth part's height is from height * 0.530 to height * 0.909. By this, we get these two parts (Left side is eye part and right side is mouth part):

(a) (b)

Fig. 7. Eye area (a), mouth area (b).

After getting the eye part, the system will start to scan the eye part from the first pixel at (0, 0) which means the pixel in the first row and first column. We have to scan the image from the first pixel because we can not assume a rough area to locate the eye position due to the rotation of face and head. Before an object is recognized as an eye, it should match some necessary conditions. First of all, when an object is found, the system will firstly get the number of pixels of it and this number should be in an acceptable range from 100 to 1200 because sometimes there will still be some noises in the image. Another condition is that the ratio of the object's length to the total length of the eye part and the ratio of the object's width to the total width should both less than a threshold value. For height, the value is 0.64 and 0.5 is for width. This condition is very useful to filter some big object such as the hair.



(a) (b)

Fig. 8. Eye area (a), detected object in eye area (b).

In the above image ,we can see there are three objects and all of them are detected (if the object is detected, it will be surrounded by a rectangle). But the biggest object on the left is hair, it will be abandoned because it does not match the rule of height ratio limit. Meanwhile rest two objects will be kept.

In the mouth part, the logic is similar with the procedure of getting eyes. The system will firstly scan the image. If an object is found and its size is bigger than a threshold value, it will be stored. And if another object is found and its length is bigger than the current one, the system will use it to replace the current object. Finally, we will get a longest object in this area like this:

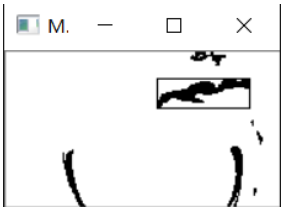
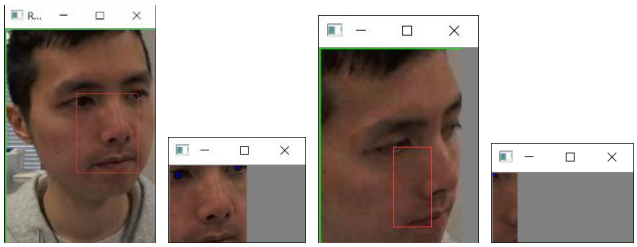


Fig. 9. Detected mouth in the mouth area.

D. Detect Nose Tip

Before starting this step, a nose area is going to be extracted by using positions of eyes and mouth in previous procedure. The nose area is a rectangle whose borders are restricted by three positions. In some cases, there is only one eye detected because of the great yaw angle of the face. Under this situation, the rectangle is extracted only by one eye position and mouth position (Left side is the nose area extracted by three nodes and right side is extracted by two nodes).



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

Fig. 10. Face image with two eyes detected (a), extracetd nose area using three nodes (b), face image with one eye detected (c), and extracetd nose area using two nodes (d).

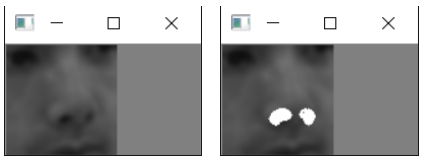
Then the nose area image will be converted into grayscale format. Then we will use the image gradient to help us to locate nose holes. The image gradient is a directional change in the intensity or color in an image. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction. Because nose holes are relatively dark comparing with other area around them, the gradient magnitude around nose holes should be larger than other area and we can get nose holes position by computing the gradient magnitude. The formula we use is as below. First step is to calculate the gradient in x direction and y direction:

$$\frac{\partial \mathrm{f}}{\partial \mathrm{x}} = [f(x+1,y) - f(x-1,y)]/2$$
$$\frac{\partial \mathrm{f}}{\partial \mathrm{y}} = [f(x,y+1) - f(x,y-1)]/2 \quad (8)$$

Then is the gradient magnitude:

$$\mathrm{mag}(x,y) = \sqrt{\left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2} \quad (9)$$

Now, we have got the gradient magnitude of the nose area. Sometimes some other area like lips or face border can also have a high value of the gradient magnitude. To filter these areas, we can use the value of gray level because nose holes are almost the darkest place in the nose area. By these two constraints, we can get nose holes (White parts in the right image are detected nose areas).



(a) (b)

Fig. 11. Nose area in grayscale format (a), detected nose area (b).

Then we can get the rough position of nose tip by calculating the average position of nose area.

IV. CALCULATION OF DISTRACTION

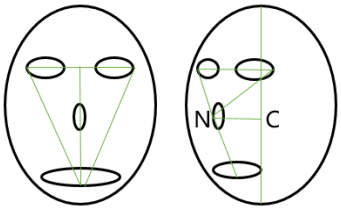
With positions of obtained facial features, we can initiate the procedure of the calculation of distraction. The calculation needs two basic parameters: the offset of the head and the offset of the face.

It is known to all that an isosceles triangle can be formed among two eyes and the mouth. If we draw a vertical line from the mouth to the line connecting two eyes. We can get the rough angle of the head by calculating the arc-tangent of that vertical line. In normal status, the angle of head is nearly 90 degrees, so the offset of the head can be get by calculating 90 minus that arc-tangent value.

$$\mathrm{OH} = 90 - \tan^{-1} K(EM) \tag{10}$$

In the above formula, K(EM) means the slope of the line from point E to point M in the left face model. And OH means the offset angle of the head.

The calculation of the offset of the face bases on the position of nose. Assume that there is a center line crossing the face from the forehead to the chin. If the face is right in front of the camera with no roll and no yaw, this center line should be able to cross the nose tip. But if the face yaw to a direction, there should be a distance A between the center line and the nose tip.



(a) (b)

Fig. 12. Face model in normal status (a), face model in left yaw status (b).

Then we define a ratio OF whose value is A dividing by the half width of the face area.

$$\mathrm{OF} = \frac{NC}{W} * 2 \tag{11}$$

In normal status, the OF should be nearly zero when the face is right in front of the camera because the nose tip is very close to the center line. And theoretically, the OF can be near 1 when the horizontal yaw angle of face reaches 90 degrees.

Now with the parameter OH and OF, the formula of detecting distraction can be preliminary given:

$$\mathrm{Dis} = \mathrm{a} * \mathrm{HA} + \mathrm{b} * \mathrm{FOF} + \mathrm{c} \tag{12}$$

In this formula, there are still three undetermined coefficients: a, b and c. These coefficients are decided by using curve fitting based on the experimental data.

$$\mathrm{Dis} = 1.286 * \mathrm{HA} + 1.048 * \mathrm{FOF} - 0.0963 \tag{13}$$

With the new formula, a threshold value with which the purpose of judging the distraction can be achieved need to be determined, which will be introduced in section VI.

V. SYSTEM ARCHITECTURE

This section describes the main workflow of the system and details of processing procedure. The figure below shows the main workflow of the system.

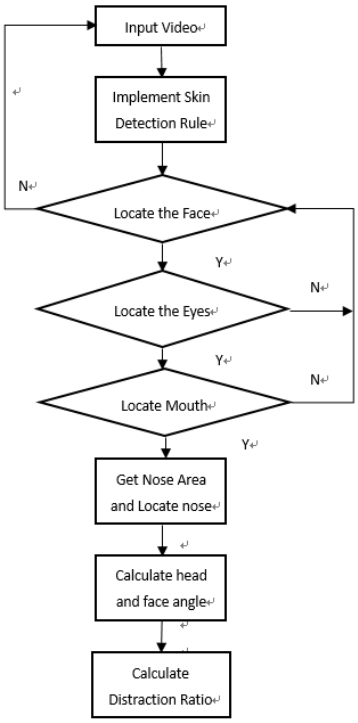


Fig. 13. WorkFlow of the System

1) The camera gets the video image as the input of the system. The skin detection rule, which consists of three sub-skin detection rules in RGB, YCrCb and HSV color spaces, is implemented on the image in order to get the skin area in the image;

2) The image is smoothed by erosion and dilation operation. Then the system will scan the image and erase those objects with inappropriate size (too big or too small). After that, the system will try to locate the biggest skin area in the image. Once the face area is found, it will be extracted as an isolated image for further process;

3) The face area will be transferred into binary format by using a threshold gray value. After this step, the skin area is turn into white while facial features such as eyes and mouth are turned into black. Then the binary image is divided into the eye part and the mouth part in which the system will try to locate eyes and mouth. The eye part can be roughly considered as the upper part of the face while the mouth part is the lower part;

4) In the eye part the system tries to locate two most suitable black objects with some rules such as appropriate position and size. Founded objects will be viewed as the rough area of the eye and the system will implement the gradient based method to locate the pupil position inside the object;

5) In the mouth part the system tries to locate a longest black object with appropriate size and location. Founded object will be viewed as the lip's position;

6) Once both positions of eyes and mouth are founded, a rough nose area using for locating the nose will be extracted by using positions of eyes and mouth. Then gradient magnitude of every pixel is get. A function then gets all pixels in the image whose gray level is lower than a given value and gradient

magnitude greater than the average magnitude. Obtained pixels can generally consist the nose hole and a nose tip point is gained by calculating the average position among these pixels;

7) Once positions of facial features are obtained. The index of distraction will be calculated using the formula in Fig.15. The status of distraction is then determined.

VI. EXPERIMENT RESULT

A series of experiments were carried out by using images collected from five students in our laboratory. Five students include 4 male and 1 female and 100 pictures are collected.

A. *Threshold value of Distraction Detection*

With the formula of calculating the index of distraction, a threshold value need to be determined to judge the status of distraction. We implement a series of experiments using some images in which the learner is staring at the edge of the monitor like the following image shows:

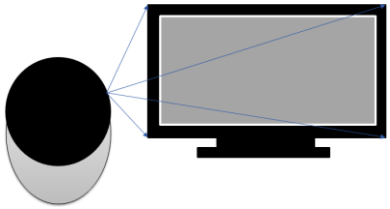


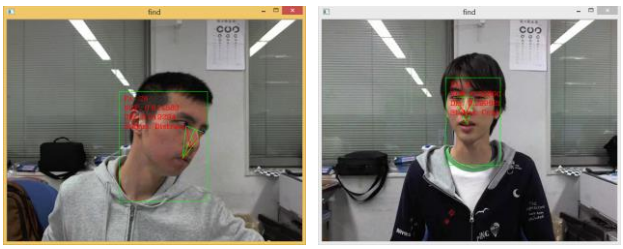
Fig. 14. The user is staring at the edge of the monitor

About 30 images was chosen to perform the test. The indexes calculated by using formula in Section IV are recorded, with which the expectation and variance is calculated:

Expectation: 0.3103 Variance: 0.0135

B. *Results of Distraction Detection*

Using the result in prior part, we can start the process of experiment. According to the experiment implemented on 100 pictures, the success rate of face detection and pose estimation is about 40%. Once the pose estimation is successful, the success rate of distraction detection is acceptable, about 90%. Here are some successful result:



(a) (b)

Fig. 15. Sample of distracted status (a), sample of concentrated status (b).

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a method of using facial feature extraction to evaluate distraction. First, the face area will be detected by using the skin detection rule. Then, the image is transferred into binary format from which the eye part and mouth part are generated for facial features extraction. In the eye part and mouth part, positions of eyes and mouth are obtained separately. In next step the position of nose tip is gained in the mouth area generated based on positions of eyes and mouth. With positions of facial features, the roll angle of the head and the yaw magnitude of the face are calculated and finally the index of distraction is gained according to which the status of distraction can be evaluated. A series of experiments are implemented to evaluate the accuracy and stability of the system. The result shows that the system can work well under some designated conditions: appropriated illumination and distance from the face to camera, no obstacle such as glasses and long hair covering facial features. However, having to admit, the experiments do expose some weaknesses of the system. The accuracy and processing speed are still not so ideal. In future, we will try some other kinds of approaches to do the facial features extraction such as training classifiers using AdaBoost and some color space methods.

REFERENCES

[1] Yatian, Chen, et al. "Research on learning-monitoring system for E-Learning." Computer Science & Education (ICCSE), 2013 8th International Conference on. IEEE, 2013.

[2] Daif, Abdullah Rady, and Mohamed Abu Rizkaa. "An enhanced model for monitoring learners' performance in a collaborative e-Learning environment." 2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE). 2013.

[3] Baptista, Christiane Meiler, Regina Melo Silveira, and Wilson Vicente Ruggiero. "MSys: a Monitoring System for E-learning Feedback and Content Fitting." Information Technology Based Higher Education and Training, 2006. ITHET'06. 7th International Conference on. IEEE, 2006.

[4] P. Peer, J. Kovac, F. Solina, "Human Skin Colour Clustering for Face Detection", EUROCON1993, Ljubljana, Slovenia, pp. 144-148, September 2003.

[5] Vatahska, Teodora, Maren Bennewitz, and Sven Behnke. "Feature-based head pose estimation from images." Humanoid Robots, 2007 7th IEEE-RAS International Conference on. IEEE, 2007.

[6] Jones, Michael, and Paul Viola. "Fast multi-view face detection." Mitsubishi Electric Research Lab TR-20003-96 3 (2003): 14.

[7] Meynet, Julien, et al. "Fast multi-view face tracking with pose estimation." Signal Processing Conference, 2008 16th European. IEEE, 2008.

[8] Garcia, Christophe, and Georgios Tziritas. "Face detection using quantized skin color regions merging and wavelet packet analysis." Multimedia, IEEE Transactions on 1.3 (1999): 264-277.

[9] bin Abdul Rahman, Nusirwan Anwar, Kit Chong Wei, and John See. "Rgb-h-cbcr skin colour model for human face detection." Faculty of Information Technology, Multimedia University (2007).

[10] Shi, Jianbo, and Carlo Tomasi. "Good features to track." Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on. IEEE, 1994.

[11] Phuong, Hoang Minh, et al. "Extraction of human facial features based on Haar feature with Adaboost and image recognition techniques." Communications and Electronics (ICCE), 2012 Fourth International Conference on. IEEE, 2012.

[12] Kim, Wongki, et al. "A feasible face pose estimation by evaluating 3D facial feature vectors from 2D features." Advanced Communication Technology (ICACT), 2013 15th International Conference on. IEEE, 2013.

[13] Li, Quanbin, Fangjiao Jiang, and Zhi Huang. "Multi-pose facial features localization based on skin color models." Image and Signal Processing (CISP), 2014 7th International Congress on. IEEE, 2014.