Emotional States Control for On-line Game Avatars

Ce Zhan, Wanqing Li, Farzad Safaei, and Philip Ogunbona
University of Wollongong
Wollongong, NSW 2522, Australia
{cz847, wanqing, farzad, philipo}@uow.edu.au

ABSTRACT

Although detailed animation has already been achieved in a number of Multi-player On-line Games (MOGs), players have to use text commands to control emotional states of avatars. Some systems have been proposed to implement a real-time automatic system facial expression recognition of players. Such systems can then be used to control avatars emotional states by driving the MOG's "animation engine" instead of text commands. Some of the challenges of such systems is the ability to detect and recognize facial components from low spatial resolution face images. In this paper a system based on an improved face detection method of Viola and Jones is proposed to serve the MOGs better. In addition a robust coarse-to-fine facial landmark localization method is proposed. The proposed system is evaluated by testing it on a database different from the training database and achieved 83% recognition rate for 4 emotional state expressions. The system is able to operate over a wider range of distance from user to camera.

General Terms

Algorithms, Experimentation, Design

Keywords

avatar control, multiplayer on-line game, facial expression recognition

1. INTRODUCTION

Multi-player On-line Games (MOGs) have become popular over the last few years, largely due to the communication, collaboration, and interactivity provided to players. Thus players are able to cooperate or compete with each other on a large scale and experience relationships often comparable to those in the real world. The attractiveness of the "real feeling" has created a whole new community of players and the supporting lucrative industry.

NetGames'07, September 19-20, 2007, Melbourne, Australia

Despite the realism of the interactivity enjoyed by players of MOGs, a comparison of the conversational aspects of the communication reveals that the interfaces are still primitive when conveying affective states of the player. Birdwhistell's linguistic analogy [2] suggests that the information conveyed by words amounts to only 20-30% of the information conveyed in a conversation. The underlying emotions conveyed by different facial expressions often make the same word have different meanings. In order to feel immersed and socially aware as in the real world, players must have an efficient method of conveying and observing changes in emotional states. Except in few MOGs where affective expressions are not supported, all other existing MOGs deal with emotions by using text commands. For example, when a player, John, types the command "/smile" while having a conversation with another player, Bruce, in a game, "John smiles at Bruce" appears on the screen. At the same time and depending on the degree of animation implemented in the MOG, John's avatar smiles or a body gesture animation is presented (like a swing of an arm). Text commands are simple and straightforward, but they do not provide an efficient and natural way to control the emotional state of the avatar.

Zhan et al. [15] proposed a real-time automatic system to recognize players' facial expressions, so that the recognition results can be used to control avatars' emotional states through the "animation engine" of the MOG. As a client application of a MOG, the system consumes less system resources and achieved a high frame rate (16 frames of 256x256 pixels per second on a PC with 2.80 GHz Intel Pentium) with an average recognition rate of 82%. However, it failed to recognize expressions accurately when the spatial resolution of input face regions is lower than 100×100. The system also exhibited decreased performance when the testers are not included in the training data.

In this paper, we improve the facial expression recognition system proposed in [15], so as to make the system deal with low resolution input and provide user-independent operation. In Section 2, an overview of the improved system is presented. Section 3 explains the essence of the object detection method proposed by Viola and Jones [13], and the modifications of the object detection method used in the proposed system for key facial component detection are presented in Section 4. A coarse-to-fine facial landmark localization scheme is demonstrated in Section 5. Section 6 describes the classification method and Section 7 presents the experimental results. Conclusions are given in Section 8.

2. SYSTEM OVERVIEW

The improved facial expression recognition system categorizes each frame of the facial video sequence into a neutral expression or one of the six basic emotions introduced by Ekman and Friesen [4], namely, *happiness*, *sadness*, *fear*, *disgust*, *surprise* and *anger*. Figure 1 shows the block diagram of the system with its five components, including, face detection, key facial component detection, landmark localization, feature extraction and classification of the expressions.

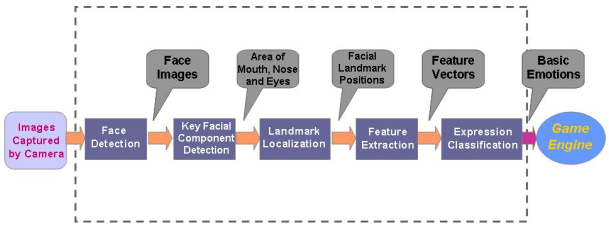


Figure 1: The architecture of the system

In the face detection and the feature extraction modules, methods similar to those adopted in [15] are used. Due to its high efficiency and detection rate, the system employs the face detection method proposed by Viola and Jones [13]; the essence of the method is introduced in Section 3. Among a number of feature extraction algorithms [5] proposed in the literature, comparative evaluation has demonstrated that Gabor filters are most discriminative for facial expressions and provide robustness against various types of noise. We apply Gabor filters only to a set of facial landmark positions rather than the whole face in order to lower the computation cost and sensitivity to illumination variations. Subsequent to feature extraction, each face image is represented by a 360-dimensional vector derived from 20 selected landmarks. Details of the feature extraction process can be found in [15].

As mentioned in Section 1, the system proposed in [15] cannot handle input face regions with the resolution lower than 100×100. Assuming a web camera with focal length of 3cm and resolution of 320×240 is used, the approximate relationship between distance of user to camera and face resolution is shown in Table 1. It is instructive to note that the corresponding distance for a 100×100 face is about 50cm. It is also noteworthy that this distance range cannot meet the requirement of a MOG, especially when wireless input devises and lower resolution cameras are used. Since Gabor filters are only applied at facial landmarks to extract facial information, the landmark localization is crucial for the entire system. In [15], fixed landmarks were used after face alignment so as to reduce computation load. However, when the scale (or spatial resolution) of the face is small, the estimation error of the approximate landmark localization increases. Experiments show that by using manually selected landmarks, the previous system works well on lower resolution input. Thus, a new coarse-to-fine method is proposed in this work to locate the facial landmarks more accurately and robustly when low resolution face images are presented to the system. As a pre-stage of landmark localization, a key facial component detection module is added to the system. In this detection module, the detection method (Viola and Jones) in [13] is modified to find the areas of mouth, nose and eyes. Another important requirement in the context of MOG is that the system must be able to handle users

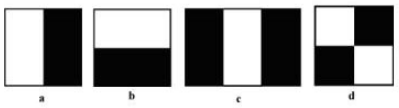


Figure 2: The rectangle features

of different gender, ages, and ethnicities. To meet this requirement, a more effective classification method, namely, support vector machine (SVM) is to categorize the different expressions. All the improvements are described in the following sections.

3. FACE DETECTION METHOD OF VIOLA AND JONES

Recently Viola and Jones [13] proposed a multi-stage object detection procedure based on AdaBoost that reduces the processing time substantially while achieving almost the same accuracy as compared to a much slower and more complex classifier. Although the method is claimed to be suitable for any object detection, it has only been demonstrated on the task of finding faces. The method achieves a fast face detection through a careful choice of features and classifier architecture.

3.1 Rectangular features

Very simple and easy to compute rectangular features, as shown in Figure 2 are used to represent the image information in a sub-window. With different sizes and positions, a sub-window in an image contains a large number of these features. The feature value in each case is simply the difference between the sum of the pixel intensities in the white section and the sum of the intensities in the black section.

3.2 Cascade of classifiers

In the training process, samples of face (named as positive samples) and non-face (named as negative samples) are rescaled to a specified sub-window size and used to train a binary classifier. In order to reduce the computation load, the binary classifier consists of multiple sub-classifiers which form a cascade. The cascade of classifiers is a degenerate decision tree where at each stage a classifier is trained to detect almost all objects of interest (faces) while rejecting a certain fraction of the non-object patterns (non-face). Each stage was trained using the fundamental boosting algorithm, AdaBoost [6], which selects and combines very simple (weak) classifiers to build a more powerful (strong) classifier. Each weak classifier use one feature from the rectangular feature pool in combination with a simple binary thresholding decision.

3.3 Detection via multi-scale scanning

In detection process, the trained cascade classifier for a certain sub-window size (size of training samples) scans over the input image at all possible locations. Then face detection is posed as classifying the pattern in the sub-window as either face or non-face. To achieve multi-scale image search, features are rescaled and thresholds are recalculated to form classifiers for different scale sub-windows.

4. KEY FACIAL COMPONENT DETECTION

	30cm	50cm	80cm	130cm
Face	165×165	100×100	65×65	45×45
Mouth	56×28	30×15	20×10	12×6
Eyes	36×18	22×11	14×7	8×4
Nose	42×42	26×26	16×16	9×9

Table 1: The approximate relationship between distance of user to camera and facial component resolution

The low computational cost of using rectangular features and the efficiency of the cascade structure suggests the adoption of "AdaBoost" method to search key facial components (nose, mouth and eyes and) within the detected face area. We note however, that the nature of objects and the detection context have changed and this leads to low detection performance when the face detection method is directly applied to fine component detection. The experimental results for mouth detection is shown in Figure 4a, and it easily seen from the plots that the size of the objects plays a significant role. Since mouth, nose and eyes are only small parts of the face, a low spatial resolution face image does not augur well for fine facial component detection. Using the same assumption as in Section 2, the approximate relationship between distance and facial component resolution is shown in Table 1. Another consideration is that the shapes of facial components vary widely when people are conveying different expressions, especially for mouths. In this situation, the mouth detector could still fail to achieve good performances even for high resolution faces. To solve these problems, we improve the "AdaBoost" detection method in several ways.

4.1 High hit rate cascade training

As introduced in section 3, each classifier in the cascade is trained with a goal to reject a certain fraction of the negative (non-object) training samples. Thus, later stage classifier faces a more difficult task since it has to reject the same fraction of negative samples among training samples which passed through all the previous stages. To handle negative samples misclassified by previous stages, classifiers in the later stages are more complex and using more subtle features. In other words, adding a stage in the cascade reduces the false positives. However, at the same time, some positives will be missed since more specified features are used and thus, the hit rate reduces with more stages. When facial components are small, the subtle information is missing and only major features are retained. They can pass through the first few stages of the trained cascade but will be rejected by more complex classifiers in the later stages of the cascade, if the cascade classifier is trained with low false positive rate.

To ensure that small scale facial components could be detected, a minimum overall hit rate is set before training. For each stage in the training, the training goal is set to achieve a high hit rate and an acceptable false positive rate. The number of features used is then increased until the target hit rate and false positives rate are met for the stage. If the overall hit rate is still greater than the minimum value, another stage is added to the cascade to reduce the overall false positive rate.

4.2 Regional scanning with a fixed classifier

In face detection, multi scale scanning is achieved by rescaling the classifier. In this process, the positions and corresponding thresholds of rectangular features selected by AdaBoost during training have to be recalculated. The discrete nature of digital image implies that discretization error arising from integer pixel coordinates cannot be avoided when the rescaling factor is fractional. This error is exacerbated when objects being detected are small.

The discretization error can be reduced by opting to resize the input image and use a fixed classifier for detection. This method is not suitable for face detection since calculating the pyramid of images for each input is time consuming. However, in the case of facial component detection, the input images are face images. Due to the structure of face, we can predict the size of face area according to the area size of facial components. Thus, during scanning, all the input face images are rescaled into one given size corresponding to the training size of facial components. Then, if the detector cannot find an object, face images are rescaled again to the size around the predicted size until the object is detected. In this way, although a fixed size detector is used, the computation of the whole image pyramid is avoided. In most cases, rescaling three times are enough to find the object.

Since in the cascade classifier training, false positive rate is somehow sacrificed to achieve a high hit rate, regional scanning is conducted to reduce false positives. Prior knowledge of face structure is used to partition the region of scanning. Area in top region of the face image is used for eye detection; the central region of the face area is used for nose detection; and mouth is searched in the lower region of the face. By regional scanning, fewer area exists that can produce false positives. It also increases efficiency since fewer features need to be computed.

4.3 Specialized classifiers

Two cascade classifiers are trained for mouth: one is trained to detect all kinds of closed mouths, and the other one is specialized for open mouth detection. During scanning, if the closed mouth detector failed to find a mouth, the open mouth detector is used. Two eyes are treated as different objects, so a right eye classifier and a left eye classifier are trained separately.

5. FACIAL LANDMARK LOCALIZATION

As mentioned in Section 2, the reliability of landmark localization has a major influence on the performance and usability of the entire system. Based on the nature of the feature, Appearance-based (e.g.[1] [12]) or Geometric-based (e.g.[3] [14]) approaches are usually applied to find accurate position of landmarks. However, most of the methods involve multiple classification steps, which is not affordable for MOGs due to the high computational cost. The robust and accurate facial component detectors we have described forms the basis of a new coarse-to-fine approach to simplify the landmark localization process, which consists of 4 steps: Estimation, Localization, Refining and Tracking.

First, approximate positions of 20 facial landmarks are estimated based on the boundary box of detected facial components. The estimation scheme is shown in Figure 3. The accuracy of the estimations can be improved by using positive training samples that are cropped tightly around eyes, mouth and nose, when training the detectors. The corresponding actual landmark is considered to localized within a $D \times D$ neighborhood of the estimated landmark, D is de-

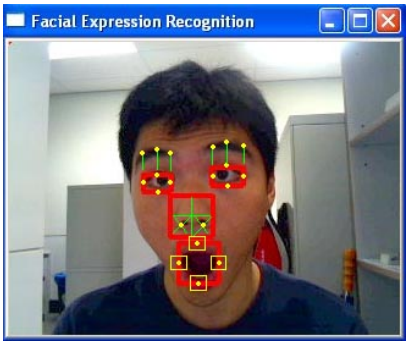


Figure 3: Landmark Estimation

termined by the size of facial components (4 neighbourhoods of mouth landmarks are indicated on Figure 3). To find the accurate landmark positions, images are converted into grey scale, thus each image can be represented by an intensity function $f(x, y)$. Within a $D \times D$ neighborhood, the location with the highest variation of intensity function $f(x, y)$ in both x and y directions is considered to be the position of a landmark. In essence localization is implemented by finding the maximum eigenvalues of local structure matrix C within neighborhoods, where

$$C = w_G(r;\sigma) * \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$$

and $w_G(r;\sigma)$ is the Gaussian filter for smoothing the matrix entries. The classic Harris [7] corner finder is applied to refine the detected landmark positions so as to achieve sub-pixel accuracy. The refining method is based on the observation that every vector from the true corner q (or a radial saddle point) to a point p located within a neighborhood of q is orthogonal to the image gradient at p. Detail of the refining process can be found in [7].

Sometimes due to out-of-plane rotations of the head, key facial components cannot be found. And there are also some cases where the true landmarks are not located in the $D \times D$ neighborhood of the estimated landmarks. With the goal of obtaining more accurate and smooth landmark positions, linear Kalman filters are employed to track landmarks detected from the above steps. The linear Kalman filter is a recursive procedure consisting of two stages: prediction and correction. During each iteration, the filter provides an optimal estimate of the current state using the current input measurement, and produces an estimate of the future state using the underlying state model. As we are interested in positional coordinates, the state vector is formulated as $\mathbf{S} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{bmatrix}^T$ and the measurement vector is formulated as $\mathbf{M} = \begin{bmatrix} x & y \end{bmatrix}^T$ where $x, \dot{x}, \ddot{x}$ are, respectively, the landmark position, velocity and acceleration in x direction. The corresponding variables in the y direction are respectively, $y, \dot{y}, \ddot{y}$. Thus, using Newtonian dynamics, the prediction process is modeled as,

$$\mathbf{S}_{k+1} = \begin{bmatrix} 1 & 0 & t & 0 & \frac{t^2}{2} & 0 \\ 0 & 1 & 0 & t & 0 & \frac{t^2}{2} \\ 0 & 0 & 1 & 0 & t & 0 \\ 0 & 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{S}_k + \begin{bmatrix} \frac{t^3}{6} \\ \frac{t^3}{6} \\ \frac{t^2}{2} \\ \frac{t^2}{2} \\ t \\ t \end{bmatrix} w_k.$$

The measurement $\mathbf{M}$, can be written as,

$$\mathbf{M}_k = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{S}_k + v_k,$$

where t is the sampling time interval(which is taken as the reciprocal of frame rate). The change rate of acceleration, w_k, is modeled as a white noise process.The measurement noise, v_k, is modeled as white noise. The random variables w_k and v_k are assumed to be independent and identically distributed (normal distribution).

Kalman filters predict landmark positions in the next frame and correct the localization results in the current frame. The prediction makes the localization process more stable when previous processing stages failed or huge error occurred. At the same time, the correction enhances the accuracy.

6. CLASSIFICATION

6.1 Support Vector Machines

Support vector machines (SVMs) [8] are a set of related supervised learning methods that try to find the biggest margin to separate different classes. Kernel functions are employed by SVMs to efficiently map input data which may not be linearly separable to a high dimensional feature space where linear methods can then be applied. Recall that there are often only subtle differences between different expressions posed by different people and this consideration has led to the poor performance of the system described in [15], especially in person-independent tests. The high discrimination ability of SVMs makes them the classifier of choice in this work. SVMs also demonstrates relatively good performance when only a modest amount of training data is used. Furthermore, since only inner products are involved in the computation of SVMs, the learning and predicting process is much faster than some traditional classifiers such as a multilayer neural network.

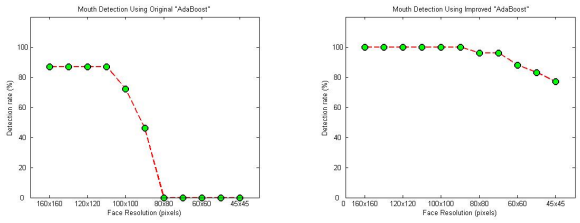
6.2 Multi-class decision making

JAFFE database [10] combined with a facial expression dataset collected in our research laboratory are used to train the SVMs. Classifiers are trained to identify Gabor coefficient vectors derived from feature extraction process, as one of the six basic emotions or a neutral expression. Since Support vector machines are binary classifiers, for 7 categories, 21 SVMs are trained to discriminate all pairs of emotions. Then, to make multi-class decisions, we combine SVM outputs by voting. For example, if one SVM makes the decision that the input is *happiness* not *sadness*, then the class *happiness* gets +1 and *sadness* gets -1. The SVMs make decisions on each pair, and thus cast votes for each category. The votes are summed together and the expression with the highest score is considered to be the final decision.

7. EXPERIMENTAL RESULTS

7.1 Facial component detection and landmarks localization

By employing the face detection method introduced in Section 3, the face detector of the proposed system can process 16 frames of 384 × 286 pixels per second on a PC with 2.80 GHz Intel Pentium, and achieve a 99.3% detection rate on the BioID face database [9]. Real-time detection examples can be seen in Figure 7.



a. Original "AdaBoost" b. Improved "AdaBoost"

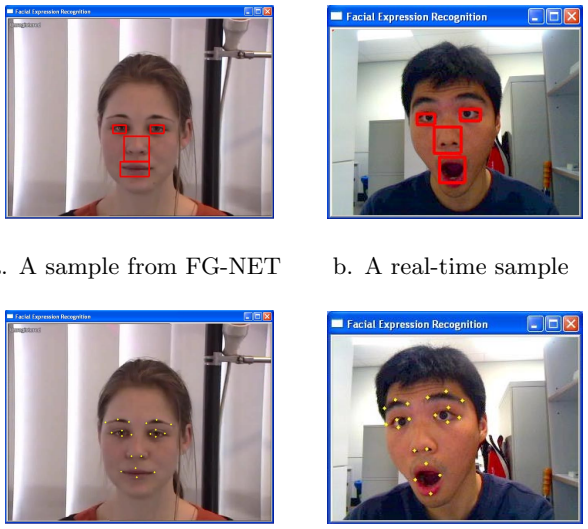
Figure 4: Mouth detection result. Both detectors are trained using the same dataset and tested by JAFFE database. All face areas of images in the database are first obtained by our face detector, then face images are down-scaled to different resolutions for testing.

Based on the improved facial component detection method, areas of eyes, mouth and nose can be detected in real-time, and the detection scheme is robust to variations in facial component's resolution and shape. The improvement gained with the new detector when compared with the original detection method is depicted in the example mouth detection result for different face resolutions presented in Figure 4b. Real-time component detection samples are shown in Figure 5 a,b.

Due to lack of benchmark video database, it is hard to evaluate the performance of landmark localization. However, the real-time test and the final recognition results reflect that the landmark localization process is robust and reliable. Examples of results with real-time samples are shown in Figure 5 c,d.

7.2 Expression recognition

We conducted a person-independent test based on the FG-NET database [11]. The database contains 399 video sequences of 6 basic emotions and a neutral expression from 18 individuals. Not all the sequences are used for testing, samples that failed to present an expression (e.g. interrupted by talking, laughing etc.) were excluded. The recognition result is presented in Table 2. The results show that *Happiness*, *Surprise* and *Neutral* were detected with relatively high accuracy while other more subtle emotions were harder to be recognized, especially the expression, *sadness*. The low recognition rate is thought to be mainly due to people conveying their emotions differently, and for more subtle expressions, the variation is wide. Some samples that were unable to be recognized, but with corresponding expressions in training database are presented in Table 4. During testing, we found that *sadness*, *anger*, *fear* and *disgust* are frequently confused with each other, however they are seldom confused with other expressions. We note even human beings sometimes have difficulty in discriminating these expressions. Thus, if these four expressions are treated as one, together with *happiness*, *surprise* and *neutral*, we can estimate user's emotional state more accurately on a higher level. Naming the new expression as *unhappy*, classification result for 4 expressions are presented in Table 3. In this way, the system is able to tell with a 83% accuracy if the user is in good mood, bad mood or just surprised.



a. A sample from FG-NET b. A real-time sample

c. A sample from FG-NET d. A real-time sample

Figure 5: Facial component detection and landmark localization

Emotion	Recognition rate
Happiness	85%
Sadness	52%
Fear	74%
Disgust	63%
Surprise	82%
Anger	69%
Neutral	80%

Table 2: Recognition results for 7 expressions classification

Emotion	Recognition rate
Happy	85%
Unhappy	86%
Surprise	82%
Neutral	80%

Table 3: Recognition results for 4 expressions classification

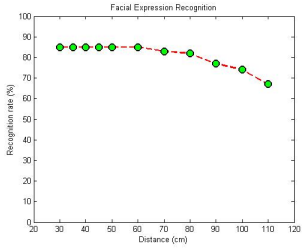


Figure 6: Recognition rates at different distances

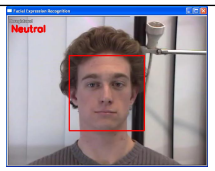
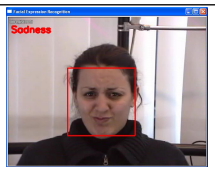
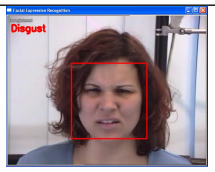
	Data in JAFFE	Data in FG-NET
Sadness		
Disgust		
Anger		

Table 4: Failed samples with corresponding expressions in training database

Tests were also conducted on the system in practical conditions, especially for low resolution input. During the test, user's expressions are classified into the four high level emotions (*happy*, *unhappy*, *surprise* and *neutral*), and a 320×240 web camera with 3cm focal length is used. The recognition result for different distance of user to camera is presented in Figure 6, and some samples are shown in Figure 7. The results show that the improved system works on the images taken from a practical range of distances from user to camera.

8. CONCLUSIONS

Research in computer vision has produced several advanced techniques for face detection, facial feature extraction, and facial landmarks localization. When integrating these algorithms to build a facial expression recognition system for a MOG, many issues were encountered. The most important problem is that people convey their emotions in so many different ways, sometimes even human are not able to recognize the emotional states of others. Since facial landmark positions are localized accurately and robustly, it is possible to use the landmarks directly to control facial actions of avatars. However, mapping 2D information onto the surface of a 3D face model is another challenging task that needs to be addressed.

9. REFERENCES

[1] G. Antonin, V. Popovici, and J. P. Thiran. Independent component analysis and support vector machine for face feature extraction. *4th International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 111–118, 2003.

[2] Birdwhistell. *Essays on Body Motion Communication*. University of Pennsylvania Press, 1970.

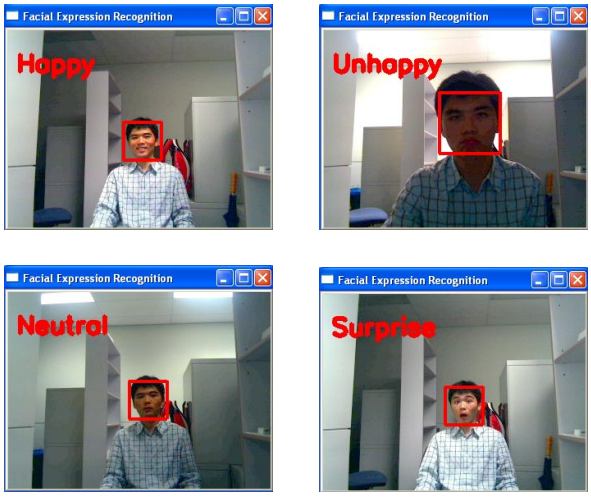


Figure 7: Recognition results for real-time test

[3] D. Colbry, G. Stockman, and A. K. Jain. Detection of anchor points for 3d face verification. *IEEE Workshop on Advanced 3D Imaging for Safety and Security*, 3(118), 2005.

[4] P. Ekman. *Emotion in the Human Face*. Cambridge University Press, 1982.

[5] B. Fasel and J. Luttin. Automatic facial expression analysis: Survey. *Pattern Recognition*, 36(1):259–275, 2003.

[6] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Computational Learning Theory: Eurocolt '95*, pages 23–37, 1995.

[7] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–151, 1998.

[8] C. Hsu and C. J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networkse*, 13(2):415–425, 2002.

[9] http://www.bioid.com/.

[10] http://www.kasrl.org/jaffe.html.

[11] http://www.mmk.ei.tum.de/ waf/fgnet/feedtum.html.

[12] Y.-S. Ryu and S.-Y. Oh. Automatic extraction of eye and mouth fields from a face image using eigenfeatures and ensemble networks. *Applied Intelligence*, 17(2):171–185, 2002.

[13] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.

[14] Z. Xue, S. Z. Li, and E. K. Teoh. Bayesian shape model for facial feature extraction and recognition. *Pattern Recognition*, 36:2819–2833, 2003.

[15] C. Zhan, W. Li, F. Safaei, and P. Ogunbona. Facial expression recognition for multiplayer online games. *Proceedings of the 3rd Australasian conference on Interactive entertainment*, pages 52–58, 2006.