International Journal of Innovative Computing, Information and Control
Volume **8**, Number **9**, September **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 6025-6036

A VISUAL CONTEXT-AWARENESS SYSTEM FOR COMPUTER ROOM CLASSES – AN AUTOMATIC ROLL-CALL SYSTEM

Jiann-Shu Lee

Department of Computer Science and Information Engineering
National University of Tainan
No. 33, Section 2, Shu-Lin Street, Tainan 70005, Taiwan
cslee@mail.nutn.edu.tw

Received May 2011; revised September 2011

Abstract. *Context-awareness is spreading into areas of computer and engineering sciences, with the goal of making systems more human-centric. Several systems have been developed for learning fields using context-aware models. However, the beneficiaries of these context-awareness systems are the learners. There are few context-aware computing systems designed for the instructor. We have developed a vision-based context awareness system to automatically monitor a student's learning status in the computer room, including roll-call and attention-sensing subsystems. In this paper, we focus on the automatic roll-call part. We utilize a two-stage face recognition scheme to identify students. The experimental results show that our system efficiently and accurately identified students via webcams, and that it is promising for automatic roll-call in computer-room classes.*

Keywords: Context-awareness, Roll-call, Computer-room classes, Chin detection

1. Introduction.

With the continuous development of computing technology, intelligent human-computer interaction has become a growing trend since the "Ubicom" (ubiquitous computing) concept proposed by Mark Weiser in the 1990s [1]. Pervasive computing, formerly only a concept, is now technologically possible with a series of novel techniques. Context-awareness, introduced by Schilit [2] in 1994, was important. Over the past two decades, context awareness has greatly evolved from its location-awareness roots to include properties such as situation, user, environmental properties. This has been possible primarily because of the use of a variety of data acquisition and processing technologies that allow designers to use a broad spectrum of contextual knowledge. These technologies convert raw contextual data into meaningful knowledge that is then used to provide awareness functionality. The conversion is generally multi-level and follows a repetitive process of information retrieval, context recognition, knowledge creation, etc.

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves [3]. There are four main types of context: location, time, identity, and environment or activity [4]. The context-awareness model allows contextual information to be collected, processed, inferred, and disseminated to spontaneous applications. It provides this interaction seamlessly, without revealing the inherent complexity required to manage the sources providing the context information. The main purpose of context-awareness systems is to adapt services or interfaces provided to their users, which is done by maintaining awareness of the surrounding situations and events. Context awareness has spread to the computer and engineering sciences, with the goal of making systems more human-centric [5].

Several educational applications use context-aware models. Zancanaro et al. [6] applied the context awareness technique to museum guides: it uses infrared sensors to detect the user's location, and then it presents multimedia information related to the fresco painting in front of which the user is standing. Jameson [7] argues for combining user modeling and context awareness, an example of which is the LISTEN system [8]: it offers audio presentations to art exhibition visitors by referring to their location and profile. Ogata and Yano [9] created a language learning system for Japanese polite expressions and personal attributes. The system prompts the user with expressions at the right level of formality for the location and participants, depending on the room the learner is in. For these and other examples [10-12], the primary entities in the context-awareness systems are learners.

There are few context-aware computing systems designed for teachers. For classroom instruction, teachers visually determine their students' learning statuses. If they find one student distracted, the teacher may call the student's name to refocus the student's attention. If most students get distracted, the teacher may try another strategy, a joke, for example. However, because of the occlusion coming from the monitors, it is difficult for the teacher to assess the students' learning status in computer classrooms. Because of the popularization of webcams, the infrastructure for visual context awareness in teaching has become mature. Nowadays, most classroom computers are equipped with webcams. Hence, it is feasible to evaluate students' learning statuses using computer vision technology. To apply context-awareness computing to computer classrooms, there are two issues of concern: student identification and concentration tracking. We have developed a vision-based context-awareness system that automatically monitors learning status, takes attendance (roll call), and senses the computer-user's attention to the monitor. In this paper, we focus on the automatic roll-call subsystem. Figure 1 shows the proposed automatic roll-call system diagram.

The roll-call system works like a video-based face-recognition system. Using the webcam mounted on top of the monitor, the system recognizes each student. This system does not interfere with student learning except in the first class, when, to link each student's

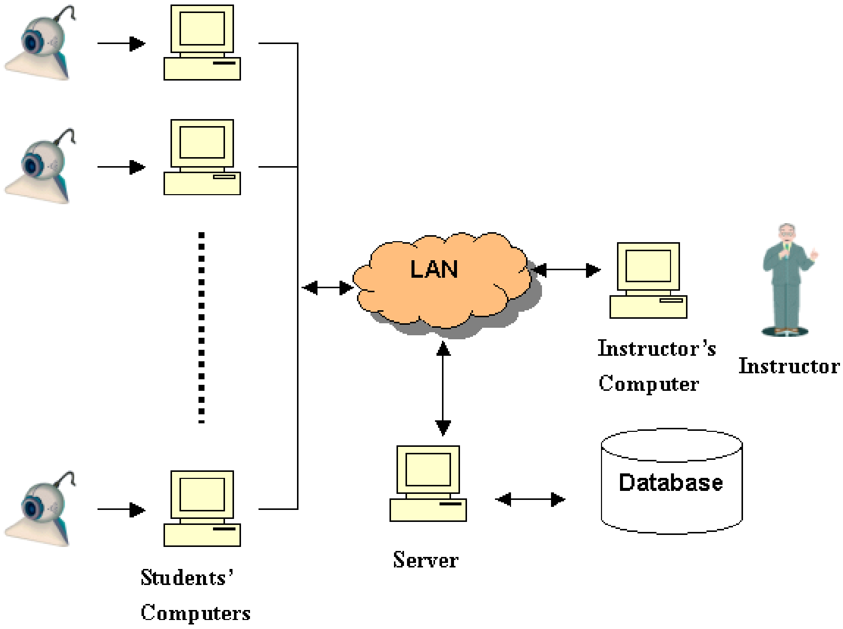


FIGURE 1. The system diagram

face with his ID, the system, triggered by the teacher, requires the student to input his student ID. Once the ID is obtained, the system starts its training phase by collecting a predefined number of frontal face images of the student. The system then switches to the recognition phase for subsequent classes.

Face recognition requires both holistic and feature information [13]. This inference can be interpreted in multiresolution aspect. Intuition tells us that, when we glance at a face, the shape, a lower-resolution feature, attracts our attention first, and then facial features that convey higher-resolution characteristics. This means that face-shape information serves as a front end for finer perception. Based on this concept, we employ a two-stage face recognition scheme – chin-shape classification and facial-feature identification [14]. This scheme was designed for static and frontal face recognition. To adapt it to meeting the requirements for the online roll-call system, we developed several new techniques such as mean-shift-based skin detection, facial-parts-based face verification, and front detection.

Skin color, identified using mean-shift clustering, segments the probable face area first. The area is then verified by examining the relationship between the eyebrows and eyes. The face area needs further examination if it is frontal. We utilize a scope-focusing process to detect the chin outline and categorize each face to a specific face shape by analyzing the chin shape. This face-shape classification step improves the face-recognition stage because it reduces the number of potential candidates. Gabor wavelets are used to extract the facial features. 2D-PCA is used to reduce feature dimensionality. Finally, a support vector machine (SVM) is used to connect the student corresponding to the face in question. Our experimental results showed that our system was highly accurate. Our "invader" test results showed that our system was highly accurate even with a high rejection rate generated by having the system scan a large number of faces ("invaders" or "strangers") not in the recognition system. These operational characteristics revealed that our system is promising for automatic roll call in computer classrooms.

This paper is structured as follows. In Section 2, how to detect the face from the grabbed frame is introduced. The chin detection is presented in Section 3. The face identification process is explained in Section 4. Section 5 presents the experimental results. Finally, we make conclusions in Section 6.

2. **Face Detection.**

Many existing face-detection methods use gray-intensity values to detect faces [15,16]. However, the skin-color features convey important information for discriminating faces from background. Because a computer classroom is bright enough to allow a color webcam to work properly, we decided to skin-color features to detect faces. The selection of an appropriate color space is required for modeling skin color. The normalized red-green space [17] is not the best choice for skin detection [18,19]. After comparing nine different color spaces for face detection, Terrillon [19] concluded that the tint-saturation-luma (TSL) space provides the best results for two kinds of Gaussian density models. We adopted the YC_bC_r space as the color space, not only because it is similar to the TSL space in terms of the separation of luminance and chrominance as well as the compactness of the chroma distribution, but also because it is perceptually uniform [20] and widely used in video compression standards.

To detect skin color, a decision rule is necessary. Creating one usually requires a metric that measures the likelihood of a given pixel color belonging to skin tone. The type of this metric is defined by the skin-color modeling method. Existing skin-color modeling methods can be divided into three categories: (a) explicitly defined skin region, (b) non-parametric skin distribution modeling, and (c) parametric skin distribution modeling. The main difficulty in achieving high recognition rates with the first method is the need to empirically find both a good color space and adequate decision rules. On the other hand, goodness-of-fit is more dependent on the distribution shape for parametric than for non-parametric skin models. Based on these considerations, we decided to use the non-parametric skin-distribution modeling method. The key idea of this method is to estimate skin-color distribution from the training data without deriving an explicit model of skin color. The result of this method sometimes is referred to as construction of a skin probability map (SPM) [21,22]. However, this method has some inherent disadvantages: a great many positive and negative training samples are needed, a great deal of storage space is required to record the corresponding SPM, and interpolation of the training data is needed. To surmount these shortcomings, we propose a clustering-based skin-detection method in which only one positive training sample is needed and an SPM record is not. Furthermore, the proposed method maintains up-to-date skin color distribution.

The preset skin-color distribution area $A_{preset\text{-}skin}$ in the C_b-C_r color space was collected from a face image captured in the computer room before our skin-color detection tests. For each computer, the grabbed frame I was segmented using the clustering algorithm. It is difficult to determine a proper cluster number; therefore, we did not adopt the k-means algorithm. Because the mean-shift algorithm [23] properly functions without needing a preset cluster number, we used that for clustering I in the C_b-C_r color space. If $\{x_i\}$ $i = 1, \ldots, n$ is the set of points of the cluster, and $w(x_i)$ the weight of pixel x_i, then the mean shift vector for point x is computed by

$$M_h(x) = \frac{\sum\limits_{x_i \in S_h(x)} w(x_i)x_i}{\sum\limits_{x_i \in S_h(x)} w(x_i)} - x \tag{1}$$

where $S_h(x)$ is the sphere centered on x, of radius h and containing n_x data points. The mean-shift vector has the direction of the gradient of the density estimate at x. The mean-shift procedure successively computes the mean-shift vector $M_h(x)$ and then translates the sphere $S_h(x)$ by $M_h(x)$. After clustering, the ith cluster is represented as C_i. The cluster C_k, viewed as the actual skin-color distribution and denoted as A_{skin}, maximizes the intersection between C_i and $A_{preset\text{-}skin}$.

$$k = \arg\max_i \{C_i \cap A_{preset\text{-}skin} : i = 1 \sim N\} \tag{2}$$

$$A_{skin} = \begin{cases} C_k & \text{if } (C_k \cap A_{preset\text{-}skin})/A_{preset\text{-}skin} > th \\ \text{No Skin} & \text{otherwise} \end{cases} \tag{3}$$

If A_{skin} is inferred to no skin, the subsequent frames are analyzed until skin color is detected. In this way, we can tune the skin-color distribution to fit the actual skin color. That is, our method adapts the skin-color model to actual conditions. The potential face area FA_p is detected using systems (4) to (7):

$$J(x,y) = \begin{cases} 1 & \text{if } (C_b^{(x,y)}, C_r^{(x,y)}) \in A_{skin} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where (x, y) is the pixel coordinate and $(C_b^{(x,y)}, C_r^{(x,y)})$ are the corresponding (C_b, C_r) values of $I(x, y)$.

$$O = \bigcup_{i=1}^{N} O_i \tag{5}$$

where O_i is the ith connected component in J.

$$m = \arg\max_i \left\{ \sum_{(x,y)\in O_i} J(x,y) : i = 1, \ldots, N \right\} \tag{6}$$

$$FA_p = O_m \tag{7}$$

$$CFA_p = holefill(FA_p) \tag{8}$$

$$IFA_p = (I \bullet B - I) \times CFA_p \tag{9}$$

Because we require only the frontal face, we have to verify whether the skin area is a human face and whether it is frontal. For human face verification, eyebrows and eyes are significant and distinct features. Therefore, human face verification focuses primarily on the relative relationship of the eyebrows and eyes. System (8) is used to get the complete potential face area by filling holes in FA_p. The input image is then converted to grayscale, in which eyes, eyebrows, nose, and mouth are represented by somewhat darker areas. This attribute combined with bottom-hat operation and CFA_p, as shown in system (9), obtains IFA_p, in which the facial features are enhanced. After thresholding IFA_p, the initial facial features are acquired (Figure 2). Subsequently, a series of filters is used to obtain the eye-and-eyebrow parts. Very small objects are filtered out by the area filter. The closing operation smoothes irregular shapes and connects neighboring objects. Objects with orientations far away from horizontal are discarded by the direction filter. The top four objects form the eye-and-eyebrow parts.

The face verification rules are based on the geometric relations between eyebrows and eyes. They are the shape similarity of the eyebrows and the distance and direction similarity between the eyes and eyebrows. First, the areas and centroids of the eyebrows

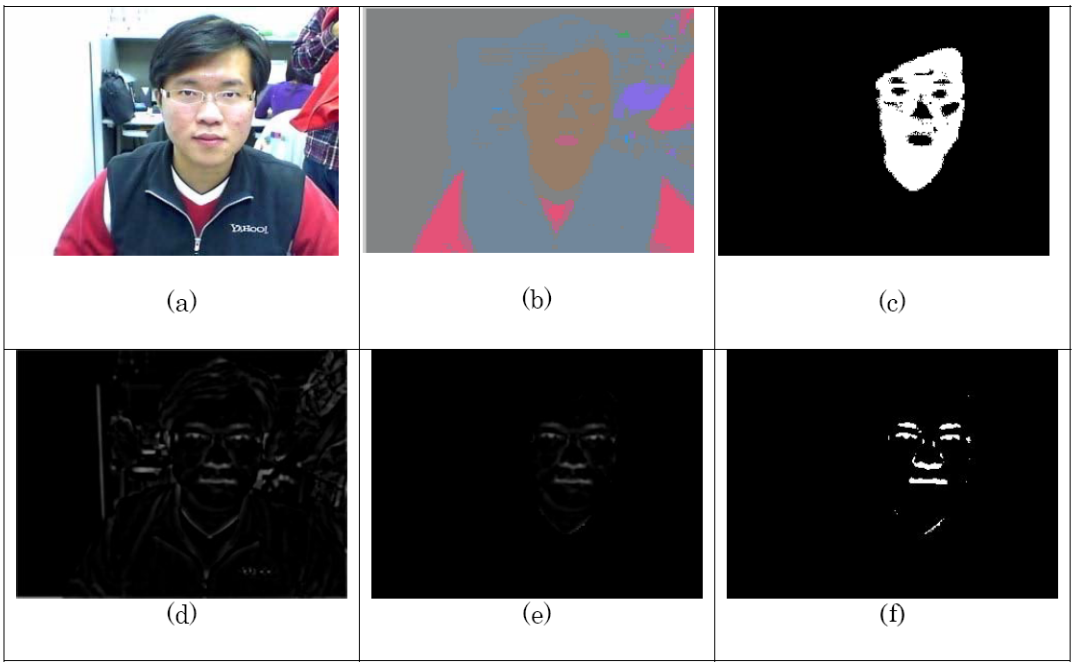


Figure 2. The result of each step in the initial facial feature extraction for an example frame. (a) The original image; (b) the clustering results using the mean-shift algorithm; (c) the FA_p; (d) the result after bottom-hat operation; (e) the result after bottom-hat operation restricted in CFA_p; (f) the thresholding result of (e).

and eyes are calculated. Let BC_L, BC_R, EC_L, and EC_R represent the centroids of the left eyebrow, right eyebrow, left eye, and right eye, respectively. For shape similarity, because the eyes may be affected by the whites of the eyes, the eyebrows are used for this comparison. The perpendicular bisector of the line connecting the left and the right eyebrow centroids is used as the symmetric axis with the unit vector $\vec{u}_\perp$ (Figure 3). The shape similarity (*SS*) between the left and the right eyebrows is calculated using systems (10) to (15), in which $\vec{X}$ represents the coordinate vector of point X, and $\tilde{p}$ is the mirror point of p with respect to $\vec{u}_\perp$.

$$\vec{u}_\perp = \frac{[BC_R^y - BC_L^y, BC_L^x - BC_R^x]}{((BC_R^y - BC_L^y)^2 + (BC_L^x - BC_R^x)^2)^{1/2}} \tag{10}$$

$$B_L = \{p \in left - eyebrow\} \tag{11}$$

$$B_R = \{q \in right - eyebrow\} \tag{12}$$

$$\vec{\tilde{p}} = 2\left(\left(\vec{p} - \vec{B}\right) \cdot \vec{u}_\perp\right) \vec{u}_\perp - \vec{p} + 2\vec{B} \tag{13}$$

$$\tilde{B}_L = \{\tilde{p}\} \tag{14}$$

$$SS = \frac{number(B_R \cap \tilde{B}_L)}{number(B_R \cup \tilde{B}_L)} \tag{15}$$

The distance similarity is quantified using the distance difference (*DD*) defined by system (16). The angle between the pair of eye-to-eyebrow vectors, *EBA*, is used to calculate direction similarity, as described in system (17). To determine suitable thresholds for *SS*, *DD*, and *EBA*, we compared the verification ratios of 30 test images (Figure 4). The selected operation points had a 100% true-positive rate and a 0% false-positive rate. Hence, we adopted thresholds of 0.4, 1, and 30 for *SS*, *DD*, and *EBA*, respectively.

$$DD = \begin{cases} \dfrac{\|EC_L - BC_L\| - \|EC_R - BC_R\|}{\|EC_R - BC_R\|}, & \text{if } \|EC_R - BC_R\| < \|EC_L - BC_L\| \\ \dfrac{\|EC_R - BC_R\| - \|EC_L - BC_L\|}{\|EC_L - BC_L\|}, & \text{if } \|EC_L - BC_L\| < \|EC_R - BC_R\| \end{cases} \tag{16}$$

$$EBA = \cos^{-1}\left(\frac{\vec{BC}_R - \vec{EC}_R}{\left\|\vec{BC}_R - \vec{EC}_R\right\|} \cdot \frac{\vec{BC}_L - \vec{EC}_L}{\left\|\vec{BC}_R - \vec{EC}_R\right\|}\right) \tag{17}$$

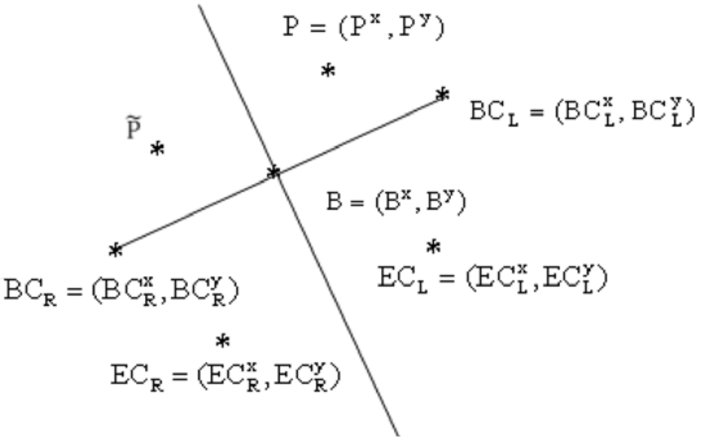


FIGURE 3. A local coordinate system formed by eyes and eyebrows

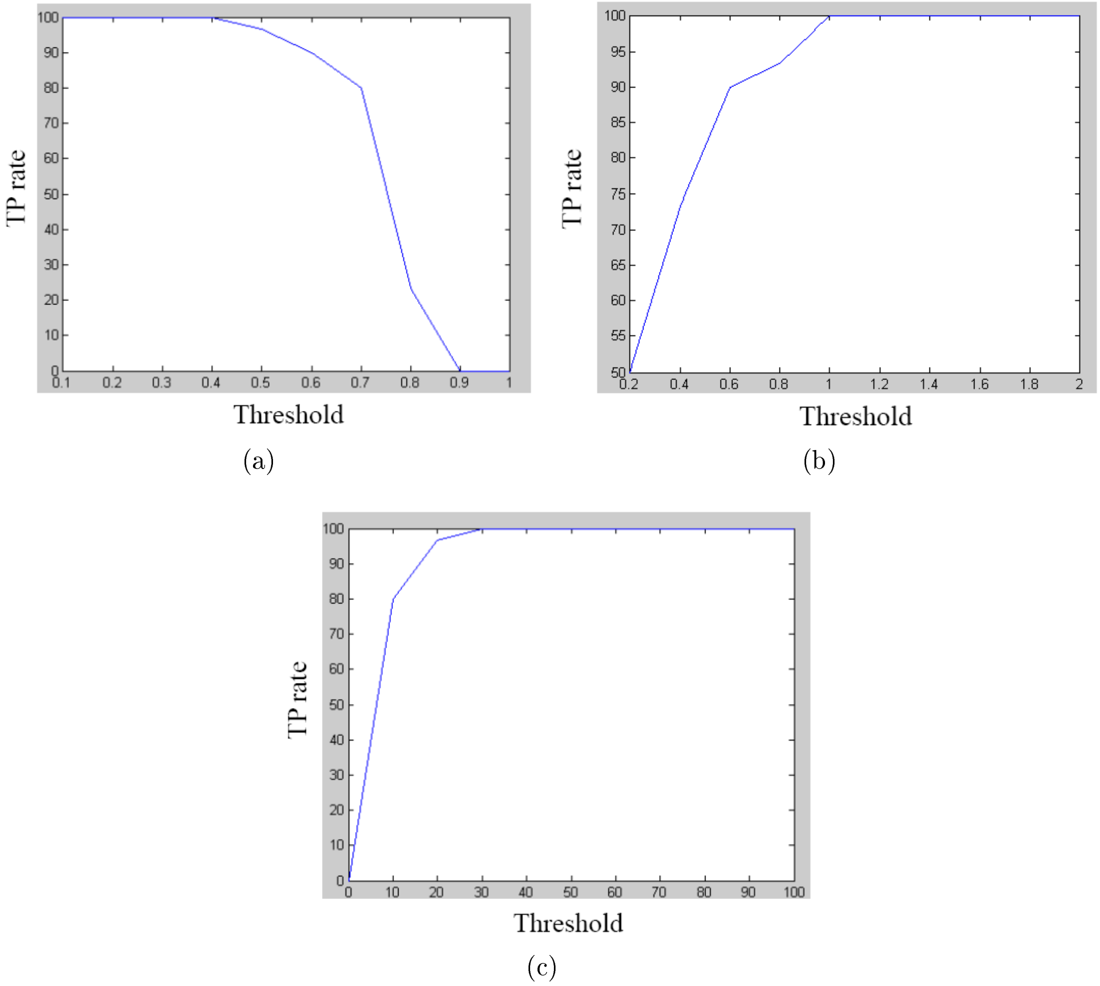


FIGURE 4. The verification ratios for various thresholds for (a) SS, (b) DD, and (c) EBA

The protocol thus far described will verify that the webcam sees a human face; however, more information is necessary to verify that the face is a frontal view. Two main non-front conditions are considered, viz. in-plane and out-plane rotations. For in-plane rotation, we correct the image by rotating Θ degrees in reference to the angle between the line connecting both EC_L and EC_R and the horizontal line (Figure 5). For out-plane rotation, we use the out-plane rotation index ($OPRI$), defined in system (18), to judge this rotation, in which LM (left margin) and RM (right margin) denote the distances from EC_L and EC_R to the left and right skin edges, respectively. If $OPRI > 1.2$, then the image is considered to have out-plane rotation and skipped.

$$OPRI = \begin{cases} \dfrac{LM}{RM}, & \text{if } LM \geq RM \\ \dfrac{RM}{LM}, & \text{if } LM < RM \end{cases} \tag{18}$$

3. **Chin Detection.** Lee et al. [14] developed a chin detection method using gray intensity information. That method was designed for static and frontal face. To adapt it to meeting the requirements for the online roll-call system, skin-tone information is exploited here. First, a horizontal line from the nose center (Figure 6(a)) is drawn to the left and

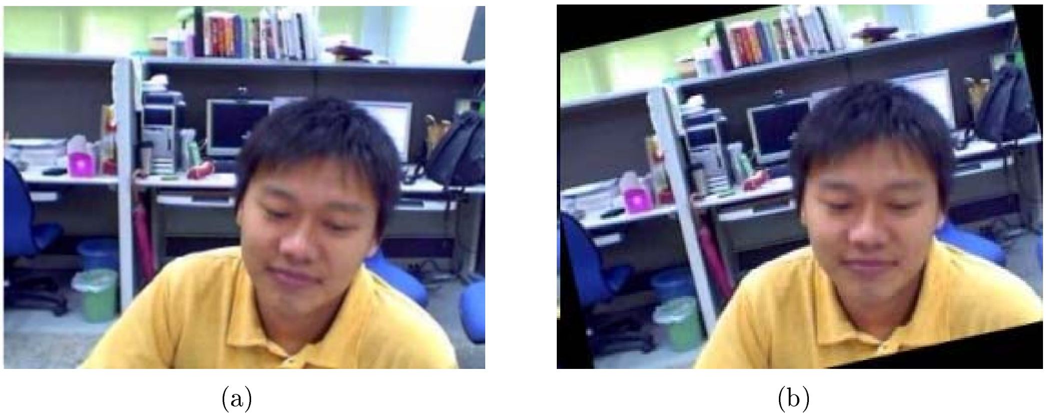
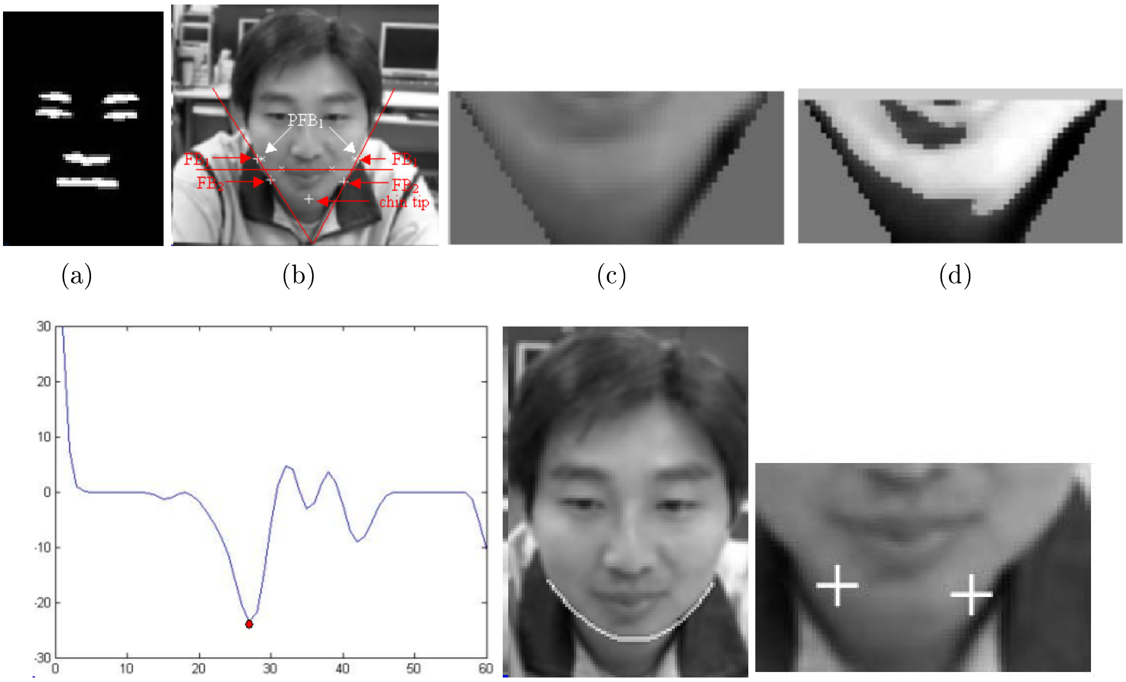


FIGURE 5. An in-plane rotation correction example: (a) the original image; (b) the rotation-corrected image

right skin edges. The two skin edge points are the first pair of preliminary face boundary points, PFB_1 (Figure 6(b)). Because skin-tone segmentation is more conservative, the actual face boundary points will be a few pixels outside PFB_1. Therefore, we fine-tune PFB_1 by detecting the edge points around PFB_1 in the grayscale image using a Gaussian first-order differential kernel. The new pair of edge points is viewed as the first pair of face boundary points, FB_1 (Figure 6(b)). The face width is thus defined as the distance between FB_1. Next, a group of pixels 1/6 the width of the face are extracted from the left corner of the mouth along the slope slanting outward at 45 degrees. The Gaussian first-order differential kernel convolves with these pixels to look for the maximum response corresponding to the face boundary point. The right boundary point is similarly obtained. These two points are denoted FB_2 (Figure 6(b)). The 4 points FB_1 and FB_2 are then extended from both the left and right sides to the intersection, and sketching the triangle containing the area below the mouth and the chin tip (Figure 6(b)). Because the chin tip is not at the lowest point of the triangle, we can limit the search range of the chin tip between the mouth and 1/2 of the face width below it. Subsequently, histogram equalization is applied to this trapezoidal area (Figures 6(c) and 6(d)). Taking the mean for every row to obtain the average brightness distribution (ABD). The chin tip is somewhere below the mouth and above the neck, where the brightness turns from bright to dark. Therefore, the minimum value of the first-order derivative of ABD will correspond to the chin tip (Figures 6(b) and 6(e)). The profile of the chin outline is drawn with a curve (Figure 6(f)) that fits the five already determined chin-boundary points (Figure 6(b)). We determine the chin features from the chin outline by using the chin classification algorithm [14]. The tangent angle of every point on the chin outline is calculated, and then the maximum and minimum values of their second-order derivatives are calculated to determine the turning points. The distance between the two turning points is defined as the chin width (Figure 6(g)).

The chin outline provides four sets of information: face width, chin curvature, face length (from the midpoint between the eyes to the tip of the chin), and chin width [14]. The Pearson correlation coefficient in system (19) is calculated to avoid highly correlated features. The chin curvature and chin width have a high correlation coefficient, which implies that the two features provide essentially the same information.

$$\gamma_{xy} = \frac{\sum (x_i - \bar{x})(x_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \tag{19}$$



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

(e) (f) (g)

Figure 6. (a) One facial feature image; (b) the five face boundary points and the bounding area used to locate the chin tip; (c) the trapezoidal area cropped from the triangle bounding area; (d) the trapezoidal area after histogram equalization; (e) the first-order derivative of *ABD* in which '•' corresponds to the chin tip; (f) the interpolated chin outline; (g) the detected chin width.

Because the chin curvature is easier to obtain, we discard the chin width. The values of the remaining three features are distance-dependent; specifically, they are dependent upon the distance between the face and the camera. To eliminate the effect of distance, we used three distance-independent features: curvature × face width, curvature × face length, and face width/face length [14]. We then use these three values to find the most effective feature combination. We adopt *k-means* to group the feature clusters and use silhouette method [24] to estimate the number of clusters represented in the data. We find that when the cluster number is 2, the combination of curvature × face length and face width/face length, the silhouette index reaches its maximum value. To automatically classify the chin outline, we adopt the back-propagation neural network as the classifier.

4. **Face Identification.** Gabor Wavelet Transform (GWT) is a common method for feature extraction. It is similar to a human's visual sense on texture features, and it contains the distinct features of direction selection and frequency selection. Hence, it is widely used in image recognition and classification. We use GWT as the feature extraction method for our webcam roll-call system. The most common Gabor wavelet filters have eight directions and five scales – thus, forty filters in total. Because the facial features are concentrated horizontally and vertically, we use only those two directions and four scales – thus, eight filters. We use the GWT by convolving the image $I(x, y)$ with the ten Gabor wavelet filters, respectively. The resulting images are then arranged to get Gabor Faces (GFs). Because the dimensionality of the GFs is large, it has to be reduced for further processing. Principal Component Analysis (PCA) is a common method to reduce dimensionality. PCA not only reduces the number of dimensions for easier calculation, but

it also eliminates the dimensions with noises. However, the disadvantage is that the GF needs to be arranged into a row vector for PCA, which means that the relations between rows in the original image may be neglected. Therefore, we used 2D-PCA to avoid this disadvantage. We then use a support vector machine (SVM) for the final human face recognition.

5. **The Experimental Results.** We used a Logitech QuickCam Pro 4000 to acquire images. The camera acquires 15 frames per second at a maximum resolution of 320 × 240. The database contained the names and student numbers of 30 college students whose presence or absence in the computer classroom was to be determined. The experiment took one minute (900 frames) of video of each student at different angles: front, side, lower head, and chin up, and at different distances. The frontal view frames were determined using the human face verification method mentioned above, and then human face recognition followed. Ten frontal face images of each student were used to train the chin-shape and face-recognition classifiers.

We tested the performance of the proposed system in two experiments. The first was a skin detection experiment on the subsequent frames based on the skin tone distribution derived from the first frame using mean-shift clustering. To find the most suitable h value for skin tone segmentation in the mean-shift algorithm, we compared the segmentation error rates (SER) defined in system (20) for different h values. The segmented skin tone area determined by hand is denoted as A_H, and the segmented area determined by mean-shift is denoted as A_{MS}. The compared results are shown in Figure 7. From this experiment, we learned that when h is 9, the minimum error rate was 11.57%. The skin-tone segmentation results for one student with various poses are shown in Figure 8. The missed skin areas are on the sides of the nose and cheeks corresponding to darker and lighter areas, respectively. Those missed areas are inside the face; thus, they can be compensated for with a filling operation that causes no problems for the subsequent analysis.

$$SER = \frac{\sum XOR(A_H, A_{MS})}{\sum A_H} \times 100 \tag{20}$$

The second experiment tested the effectiveness of the roll-call system. From a collection of 20 front images of each student in one class with 30 students, 10 images were randomly selected for training, and 10 for testing. That is, we have 300 images for identification.

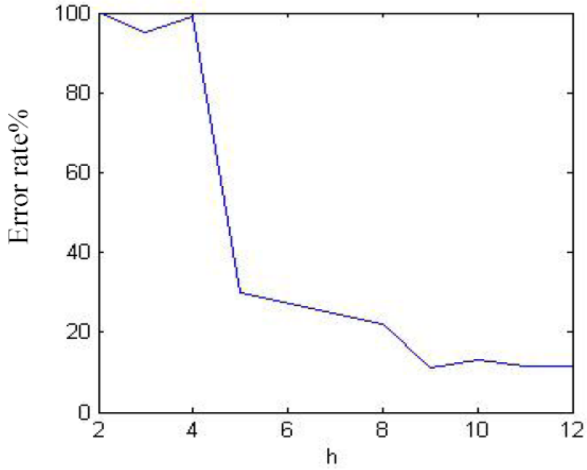


Figure 7. The skin segmentation errors for different mean shift radiuses

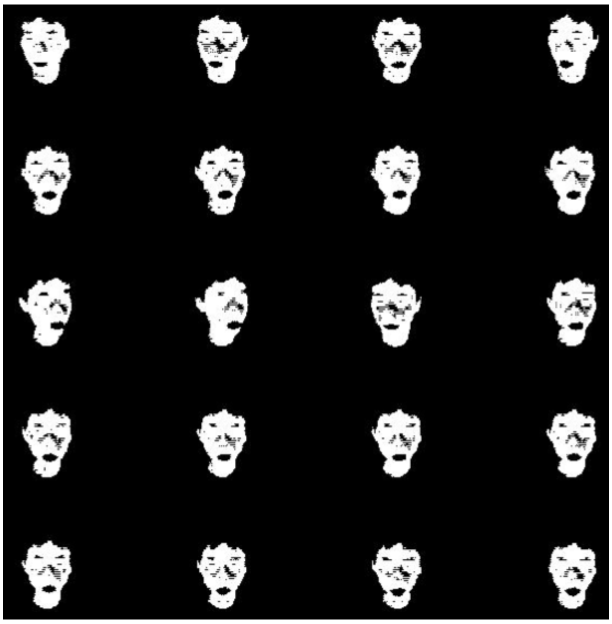


FIGURE 8. Some skin segmentation results for one subject with various poses

We found that our system can correctly recognize all students. We also found that chin-outline information indeed improved the recognition performance. Some students were mismatched when the chin-shape classification stage was omitted and only GFs information was used. Using the chin-shape classification stage, however, eliminated this error. We also compared the performance of our method with that of the method proposed by Ekenel et al. [25], which we chose because it also uses video sequences as input. Both methods were required to recognize the same test video sequences. The recognition rate was 95% for their method [25]. The comparison result shows that our system can provide superior performance.

6. Conclusions.

In this paper, we proposed and tested an automatic roll-call system to help teachers take attendance in computer classrooms. The system consists of a two-stage student identification scheme – chin-shape classification and holistic face-recognition, inspired by how humans perceive faces. First, the system uses a mean-shift based skin-color modeling to segment the probable face areas. These areas are then verified by examining the relationship between the eyebrows and eyes and both in-plane and out-plane rotations. The face is then classified as a specific face shape by analyzing the chin shape. This face-shape classification step improves the face-recognition stage because it reduces the candidate number of possible faces. The system uses Gabor wavelets to extract the facial features, and 2D-PCA is applied to reduce feature dimensionality. Finally, the system uses SVM to identify the corresponding student for the face being processed. Our experimental results showed that our system correctly identified 100% of the tested students. In addition, from the experimental results we also note that face-shape information improved the system's face-recognition performance. Taken together, all these findings strongly indicate that our system is a good candidate for an automatic roll-call application in computer classrooms.

Acknowledgment. This work was partially supported by grant NSC 96-2221-E-024-016 from the National Science Council, Taiwan. The author would like to thank Mr. Seng-Fong Lin for his assistance in designing the code.

REFERENCES

[1] M. Weiser, The computers for the twenty-first century, *Scientific American*, vol.265, no.3, pp.94-104, 1991.

[2] W. N. Schilit, *A System Architecture for Context Aware Mobile Computing*, Ph.D. Thisis, Columbia University, 1995.

[3] A. K. Dey and G. D. Abowd, Towards a better understanding of context and context-awareness, *GVU Technical Report GIT-GVU-99-22*, College of Computing, Georgia Institute of Technology, 1999.

[4] T. Gross and M. Specht, Awareness in context-aware information systems, *Proc. of Mensch & Computer*, Bonn, 2001.

[5] A. Godbole and W. W. Smari, A methodology and design process for system generated user interruption based on context, preferences, and situation awareness, *Proc. of the IEEE International Conference on Information Reuse and Integration*, pp.16-18, 2006.

[6] M. Zancanaro, O. Stock and I. Alfaro, Mobile cinematic presentations in a museum guide, in *Book of Abstracts, Learning and Skills Development Agency*, London, 2003.

[7] A. Jameson, Modelling both the context and the user, *Personal Technologies*, vol.5, no.1, pp.1-4, 2001.

[8] A. Zimmerman, A. Lorenz and M. Specht, User modelling in adaptive audio-augmented museum environments, *Proc. of the 9th International Conference User Modelling*, pp.403-407, 2003.

[9] H. Ogata and Y. Yano, Context-aware support for computer-supported ubiquitous learning, *Proc. of IEEE International Workshop on Wireless and Mobile Technologies in Education*, pp.27-34, 2004.

[10] J. Kolari, Kontti – Context-aware mobile portal, *ERCIM News*, vol.54, pp.14, 2003.

[11] D. Chalmers, M. Sloman and N. Dulay, Contextual mediation enables appropriate data selection, *ERCIM News*, vol.54, pp.15-16, 2003.

[12] Y. Cui and S. Bull, Context and learner modelling for the mobile foreign language learner, *System*, vol.33, no.2, pp.353-367, 2005.

[13] V. Bruce, P. J. B. Hancock and A. M. Burton, Human face recognition and identification, *Face Recognition: From Theory to Applications*, pp.51-72, 1998.

[14] J.-S. Lee and S.-F. Lin, A hierarchical face recognition scheme, *International Journal of Innovative Computing, Information and Control*, vol.6, no.12, pp.5439-5450, 2010.

[15] H. A. Rowley, S. Bluja and T. Kanade, Neural network based face detection, *IEEE Trans. on Pattern. Analysis Mach. Int.*, vol.20, no.1, pp.39-51, 1998.

[16] M. C. Burl, T. K. Leung and P. Perona, Face localization via shape statistics, *Proc. of the 1st International Workshop on Face and Gesture Recognition*, Zurich, Switzerland, 1995.

[17] L. M. Bergasa, A. Gardel, M. A. Sotelo and L. Boquete, Unsupervised and adaptive Gaussian skin-color model, *Image and Vision Computing*, vol.18, pp.987-1003, 2001.

[18] E. Saber and A. M. Tekalp, Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions, *Pattern Recognition Letters*, vol.19, pp.669-680, 1998.

[19] J. C. Terrillon, M. N. Shirazi, H. Fukamachi and S. Akamatsu, Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images, *Proc. of IEEE Int'l Conf. Face and Gesture Recognition*, pp.54-61, 2000.

[20] C. A. Poynton, *A Technical Introduction to Digital Video*, John Wiley & Sons, 1996.

[21] J. Brand and J. Mason, A comparative assessment of three approaches to pixel-level human skin-detection, *Proc. of the International Conference on Pattern Recognition*, vol.1, pp.1056-1059, 2000.

[22] G. Gomez, On selecting colour components for skin detection, *Proc. of the ICPR*, vol.2, pp.961-964, 2000.

[23] D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.24, pp.603-619, 2002.

[24] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *J. Comp. App. Math*, vol.20, pp.53-65, 1987.

[25] H. K. Ekenel, J. Stallkamp, H. Gao, M. Fischer and R. Stiefelhagen, Face recognition for smart interactions, *Proc. of IEEE International Conference on Multimedia and Expo*, pp.1007-1010, 2007.