Fuzzy System Learned Through Fuzzy Clustering and Support Vector Machine for Human Skin Color Segmentation

Chia-Feng Juang, *Member, IEEE*, Shih-Hsuan Chiu, and Shen-Jie Shiu

***Abstract*—This paper proposes a Fuzzy System learned through Fuzzy Clustering and Support Vector Machine (FS-FCSVM). The FS-FCSVM is a fuzzy system constructed by fuzzy if-then rules with fuzzy singletons in the consequence. The structure of FS-FCSVM is constructed by fuzzy clustering on the input data, which helps to reduce the number of rules. Parameters in FS-FCSVM are learned through a support vector machine (SVM) for the purpose of achieving higher generalization ability. In contrast to nonlinear kernel-based SVM or some other fuzzy systems with a support vector learning mechanism, both the number of parameters/rules in FS-FCSVM and the computation time are much smaller. FS-FCSVM is applied to skin color segmentation. For color information representation, different types of features based on scaled hue and saturation color space are used. Comparisons with a fuzzy neural network, the Gaussian kernel SVM, and mixture of Gaussian classifiers are performed to show the advantage of FS-FCSVM.**

***Index Terms*—Color segmentation, fuzzy clustering, fuzzy neural network (FNN), mixture of Gaussian classifier (MGC), structure learning.**

I. Introduction

COLOR image segmentation can be viewed as a classification problem based on color and spatial features. Color information appearing in an image is an important feature for human vision to cluster desired objects. Color image segmentation [1], [2] is a critical and essential component of image analysis and pattern recognition; moreover, it is one of the most difficult tasks in color image processing. It also determines the quality of the final result of analysis. For skin segmentation, the main advantage of using color information is that segmentation is independent of its size and position within the image. Generally, approaches for skin color segmentation include the following categories: 1) histograms [3]–[5]; 2) statistical learning [6]–[8]; and 3) neural networks [8]–[11]. For histogram skin classifiers (HSC) [4], the main drawback is that a huge amount of training data is usually required. Of the statistical learning approaches, the most widely used is the mixture of Gaussian classifier (MGC) trained by the expectation–maximization (EM) algorithm. MGC trained by the EM algorithm may generalize well based on a smaller amount of training data than HSC. However, the training performance of the EM algorithm may not be good enough. With neural network approaches, the training performance is usually good. However, the generalization ability may be poor due to overtraining.

The fuzzy neural network (FNN) [13], [14] and support vector machine (SVM) [15]–[17] are two types of computational intelligence techniques with high capacity and efficiency. For FNN, the parameters are learned by minimizing only the training error (or empirical risk), which does not imply a small test error. Thus, generalization performance may be poor when the FNN is overtrained. The SVM is a new and powerful network, whose formulation of learning is based on the principle of structural risk minimization. Instead of minimizing an objective function based on the training, the SVM attempts to minimize a bound on the generalization error. SVMs have gained wide acceptance due to their high generalization ability over a wide range of applications and to their better performance than other traditional learning machines, such as radial basis function networks [18] or backpropagation neural networks [19].

The learning of fuzzy rules based on SVM has been previously proposed [19], [20]. In these studies, fuzzy rules are generated by a Gaussian kernel SVM, where a support vector (SV) is regarded as a fuzzy rule. Although fuzzy rules are automatically generated, the number of fuzzy rules is equal to the number of SVs, which is usually very large, particularly for the skin-color segmentation problem studied in this paper. For this problem, we propose a Fuzzy System learned through Fuzzy Clustering and Support Vector Machine (FS-FCSVM).

FS-FCSVM is a fuzzy system whose structure is generated via fuzzy clustering on the input data, and the consequent part parameters are trained by SVM to improve the system generalization performance. It is applied to skin color segmentation, where the scaled hue and saturation (hS) values are used as color features and fed as classifier inputs. Advantages of FS-FCSVM are verified through comparisons with an FNN and SVM on the segmentation problem.

This paper is organized as follows. In Section II, the color feature used for skin color segmentation is described. Structure of the FS-FCSVM is introduced in Section III. Structure and parameter learning of FS-FCSVM are introduced in Sections IV

Manuscript received February 15, 2006; revised August 14, 2006. This work was supported by the National Science Council of Taiwan, R.O.C., under Grant NSC-95-2218-E-005-003. This paper was recommended by Associate Editor J. Wu.

C.-F. Juang and S.-J. Shiu are with the Department of Electrical Engineering, National Chung Hsing University, Taichung 402, Taiwan, R.O.C. (e-mail: cfjuang@dragon.nchu.edu.tw).

S.-H. Chiu is with Realtek Semiconductor Corporation, Hsinchu 300, Taiwan, R.O.C.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2007.904579

and V, respectively. Experimental results are presented in Section VI. Finally, conclusions are drawn in Section VII.

II. Color Feature for Segmentation

Three numerical components are necessary to describe a color, as human vision is based on three types of color photoreceptor cone cells. Color is perceived by humans as combinations of red (R), green (G), and blue (B) which are usually called three primary colors [21]. We can derive other kinds of color spaces by using either linear or nonlinear transformations. Linear transformations include YIQ (Y stands for the luma component and I and Q are the chrominance components) and YUV (U and V are the chrominance components), and nonlinear transformations include YCrCb, hue, saturation, and value (HSV), normalized RGB, etc. [2].

In this paper, color-based skin segmentation is studied. Image segmentation refers to partitioning an image into different regions that are homogeneous with respect to some image features. Several color models have been proposed for skin color segmentation, such as normalized RGB, YCrCb, and HSV [22]. These color models are used because performing the segmentation process in these spaces is not so sensitive to changes in illumination [23]–[27]. In [23], normalized rg color feature was used for skin color modeling. In [24] and [25], different color features were tested and normalized color spaces were suggested for skin color segmentation and face detection. In [26], CrCb color feature was used for face detection. In [27], an adaptive color space switching method for face tracking was proposed. A test using five 2-D color features (RG, rg, HS, YQ, and CbCr) showed that switching between the RG and HS color features resulted in increased tracking performance. In [28], the HSV color space was found to be better suited than normalized rg color space for estimation and prediction of skin-color distribution evolution in image sequences. The studies above show that no color spaces can predominate for all kinds of color images and segmentation methods. Selecting the best color space is still one of the difficulties in color image segmentation [22]. In this paper, choice of the HSV color space was motivated by the results in [27] and [28].

The HSV color space separates the color information of an image from its intensity information. RGB pixel values can be transformed to the corresponding HSV entries using the following three equations [21], [22]:

$$H1 = \cos^{-1}\frac{0.5\left[(R-G)+(R-B)\right]}{\sqrt{(R-G)^2+(R-B)(G-B)}}$$
$$H = H1, \qquad \text{if } B \le G$$
$$H = 360 - H1, \qquad \text{if } B > G \tag{1}$$
$$S = \frac{\text{Max}(R,G,B) - \text{Min}(R,G,B)}{\text{Max}(R,G,B)} \tag{2}$$
$$V = \frac{\text{Max}(R,G,B)}{255} \tag{3}$$

where $0 \le R, G, B \le 255$. There are two advantages of the HSV color space: 1) the brightness or lightness component is irrelevant to chromatic information of the images and 2) the chromatic component consisting of hue and saturation is intuitive. To eliminate the effect of lighting on the segmentation performance, only HS color space is used in this paper. The HS values are fed as inputs to FS-FCSVM. The original ranges of H and S are [0, 360] and [0, 1], respectively. Before feeding HS values to a classifier, the range of H is scaled to [0, 1], so that its range is the same as S value. This scaled HS color values are denoted as hS.

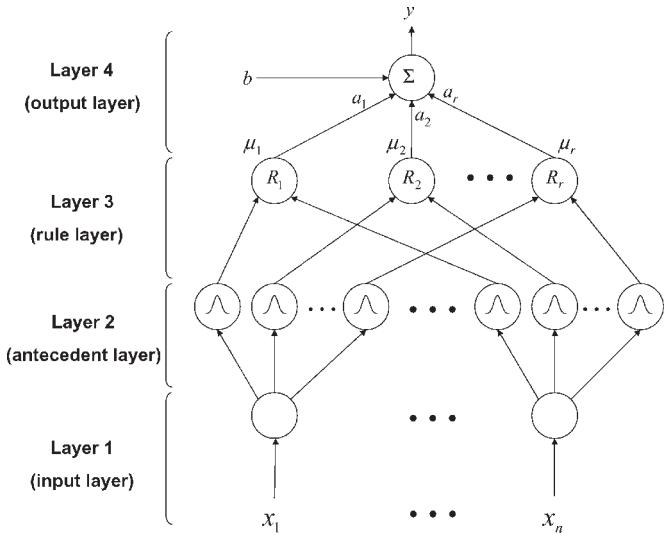

Fig. 1. Structure of the FS-FCSVM.

III. Structure of FS-FCSVM

The structure of FS-FCSVM is shown in Fig. 1. Each rule in FS-FCSVM has the following form:

Rule k :

$$\text{IF } x_1 \text{ is } A_{k1} \text{ And}, \ldots, \text{And } x_n \text{ is } A_{kn} \text{ Then } y' \text{ is } a_k \tag{4}$$

where a_k is a fuzzy singleton, and A_{kj} is a fuzzy set. The model of this four-layered network is described as follows.

In Layer 1, each node corresponds to one input variable and directly transmits input values to the next layer, thus requires no computation. The labeled training data set S is supposed to be as follows:

$$S = \left\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_N, y_N)\right\}. \tag{5}$$

In Layer 2, each node corresponds to one fuzzy set and calculates a membership value. In this layer, fuzzy set A_{kj} is employed with the following Gaussian membership function:

$$M_{kj}(x_j) = \exp\left\{-\left(\frac{(x_j - m_{kj})^2}{\sigma_k^2}\right)\right\} \tag{6}$$

where m_{kj} and σ_k denote the center and width of the fuzzy set, respectively. This Gaussian membership function has been widely used in neural fuzzy systems [13]. Here, for each fuzzy set connected to node k, the width is the same and is denoted as σ_k.

In Layer 3, each node represents a fuzzy logic rule and performs antecedent matching of this rule using the following AND operation:

$$\mu_k(\vec{x}) = \prod_{j=1}^{n} M_{kj}(x_j) = \exp\left\{-\sum_{j=1}^{n}\left[\frac{(x_j - m_{kj})^2}{\sigma_k^2}\right]\right\} = \exp\left\{-\frac{\| \vec{x} - \vec{m}_k \|^2}{\sigma_k^2}\right\} \quad (7)$$

where $\vec{x} = [x_1, \ldots, x_n]^T$, and $\vec{m}_k = [m_{k1}, \ldots, m_{kn}]^T$.

In Layer 4, each node corresponds to one output variable. Here, the simple weighted sum is used for the defuzzification operation. The node integrates all the actions recommended by Layer 3 plus a bias. Thus, the output can be written as

$$y' = \sum_{k=1}^{r} a_k \cdot \exp\left\{-\frac{\| \vec{x} - \vec{m}_k \|^2}{\sigma_k^2}\right\} + b = \sum_{k=1}^{r} a_k \cdot \mu_k(\vec{x}) + b. \quad (8)$$

A bias term b is included in this layer to account for the bias term in the linear SVM which is used to solve the free parameters in this layer.

FS-FCSVM is proposed in order to combine advantages and eliminate disadvantages of both FNN and SVM, as analyzed below. In FNN, the learning is based on empirical risk minimization (or minimizing only the training error), so its generalization ability is not as good as SVM. However, the number of parameters (or rules) in an FNN is usually smaller than in an SVM. In SVM, the learning is based on structural risk minimization (or test error minimization), so its generalization ability is usually higher than FNN, but the number of parameters (or number of SVs) in a SVM is usually much larger than in an FNN. In contrast to FNN and SVM, FS-FCSVM is trained to characterize with high generalization ability and small network size at the same time. To achieve this objective, structure and parameters in FS-FCSVM are trained by fuzzy clustering and SVM, respectively. Details of the learning processes are introduced in the following two sections.

IV. Structure Learning

The objective of structure learning is the partitioning of the input space, which influences the number of fuzzy rules generated. The concept of the clustering algorithm in [14] is used, except that the cluster alignment process is discarded. The used clustering algorithm is a one-pass algorithm that can quickly determine the antecedent part. The firing strength $\mu_k(\vec{x})$ in (7) is used as the criterion to decide if a new fuzzy rule will be generated. For the first incoming data $\vec{x}\ (0)$, a new fuzzy rule is generated, with the center and width of Gaussian membership function assigned as

$$m_{1j} = x_j(0), \qquad \text{for } j = 1, \ldots, n, \text{ and } \sigma_1 = \sigma_{\text{init}} \quad (9)$$

where σ_{init} is a prespecified value that determines the initial width of the first cluster. For the succeeding incoming data $\vec{x}\ (t)$, find

$$K = \arg \max_{1 \le k \le r(t)} \mu_k\left(\vec{x}\ (t)\right) \quad (10)$$

where $r(t)$ is the number of existing rules at time t. If $\mu_K \le \mu_{\text{th}}$, then a new rule is generated, where $\mu_{\text{th}} \in (0, 1)$ is a prespecified threshold. A larger value of F_{in} generates a higher number of rules. In general, a higher number of rules results in a smaller training error. However, when the rule number is too large, the improvement is limited. In this paper, parameter μ_{th} is heuristically chosen according to the above rule of thumb. Once a new rule is generated, the next step is to assign centers and widths of the corresponding membership functions. Here, these values are assigned by

$$\vec{m}_{(r(t)+1)} = \vec{x}\ (t) \quad (11)$$

$$\sigma_{(r(t)+1)} = 0.5 \cdot \frac{\| \vec{x} - \vec{m}_K \|^2}{\sigma_K^2} \quad (12)$$

where the width σ is assigned to be half the distance between the input variable and cluster K. Equation (12) shows that the width of each fuzzy set in the same rule (cluster) is identical and that this is not the case for fuzzy sets in different fuzzy rules. In [14], the widths of each cluster are further tuned by neural learning. That is, the widths of each fuzzy set in the same rule are different after learning, totally resulting in nr widths. Therefore, cluster alignment is performed to reduce the number of fuzzy sets (parameters) in the antecedent part in [14]. In FS-FCSVM, only the consequent part parameters are tuned by SVM, and r widths are recorded for r rules after learning. Thus, the cluster alignment algorithm in [14] is not used in FS-FCSVM.

V. Parameter Learning

The structure of FS-FCSVM is learned via fuzzy clustering on the input data. As to the consequent part parameters, i.e., the a_k values, they are determined by SVM. In this section, basic concepts of classification by SVM [17] are described followed by the consequent parameter learning by SVM.

A. Classification by SVM

Suppose we are given a set S of the labeled training set, $S = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_N, y_N)\}$, where $\vec{x}_i \in \mathbb{R}^n$, and $y_i \in \{+1, -1\}$. The SVM learning approach attempts to find the hyperplane $\vec{w}^T\vec{x} + b = 0$ defined by $\vec{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$, such that we can separate the data according to the following decision function:

$$f(\vec{x}) = \text{sign}(\vec{w}^T\vec{x} + b). \quad (13)$$

Considering that the training data are linearly nonseparable, the goal of the SVM is to find an optimal hyperplane such that

$$y_i\left(\vec{w}^T\vec{x}_i+b\right)\geq 1-\xi_i,\qquad i=1,\ldots,N \tag{14}$$

where $\xi_i\geq 0$ is a slack variable, with $\xi_i>1$ denoting data misclassification. Now, the goal of the SVM is to find a separation hyperplane for which the misclassification error can be minimized while maximizing the margin of separation. Finding an optimal hyperplane requires solving the following constrained optimization problem:

$$\begin{aligned}&\text{Min}_{w,\xi}\ \frac{1}{2}\vec{w}^T\vec{w}+C\sum_{i=1}^{N}\xi_i\\&\text{Subject to } y_i\left(\vec{w}^T\vec{x}_i+b\right)\geq 1-\xi_i\end{aligned} \tag{15}$$

where C is a user-defined positive parameter, and $\sum\xi_i$ is an upper bound on the number of training errors. Parameter C controls the tradeoff between error and margin, and a larger value of C corresponds to assigning a higher penalty to errors. By forming the Lagrangian, the primal problem in (15) can be converted into the following equivalent dual problem [17]:

$$\begin{aligned}&\text{Max}_\alpha\ L(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j\langle\vec{x}_i,\vec{x}_j\rangle\\&\text{Subject to }\sum_{i=1}^{N}\alpha_i y_i=0,\qquad 0\leq\alpha_i\leq C\end{aligned} \tag{16}$$

where α is Lagrange multiplier and

$$\vec{w}=\sum_{i=1}^{N}\alpha_i y_i\ \vec{x}_i. \tag{17}$$

After solving (16), we have the final hyperplane decision function as follows:

$$\begin{aligned}f(\vec{x})&=\text{sign}(\vec{w}^T\vec{x}+b)\\&=\text{sign}\left(\sum_{i=1}^{N}y_i\alpha_i\langle\vec{x},\vec{x}_i\rangle+b\right)\\&=\text{sign}\left(\sum_{i\in\text{SV}}y_i\alpha_i\langle\vec{x},\vec{x}_i\rangle+b\right)\end{aligned} \tag{18}$$

where the training samples for which $\alpha_i\neq 0$ are SVs. Detailed derivation process can be found in [17].

The above linear SVM can be readily extended to a nonlinear classifier by using a nonlinear operator Φ to map the input data into a higher dimensional feature space. When $\vec{x}$ is replaced by its mapping in the feature space Φ, the dual optimal problem in (16) becomes

$$\begin{aligned}&\text{Max}_\alpha\ \text{L}(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j\Phi(\vec{x}_i)\Phi(\vec{x}_j)\\&\text{Subject to }\sum_{i=1}^{N}\alpha_i y_i=0,\qquad 0\leq\alpha_i\leq C.\end{aligned} \tag{19}$$

After solving (19), the decision function can be written as

$$\begin{aligned}f(\vec{x})&=\text{sign}\left(\sum_{i=1}^{N}y_i\alpha_i\left\langle\Phi(\vec{x}),\Phi(\vec{x}_i)\right\rangle+b\right)\\&=\text{sign}\left(\sum_{i=1}^{N}y_i\alpha_i K(\vec{x},\vec{x}_i)+b\right)\\&=\text{sign}\left(\sum_{i\in\text{SV}}y_i\alpha_i K(\vec{x},\vec{x}_i)+b\right)\\&=\text{sign}\left(\sum_{i\in\text{SV}}w_i' K(\vec{x},\vec{x}_i)+b\right)\end{aligned} \tag{20}$$

where $w_i'=\alpha_i y_i$. Equation (20) shows that the number of w_i' parameters is equal to the number of SVs, which is usually very large. There are many kernel functions that can be used, such as the polynomial kernel and the Gaussian kernel with $K(\vec{x},\vec{x}_i)=\exp(-\|\ \vec{x}-\vec{x}_i\ \|^2/\sigma^2)$, but all of them must satisfy Mercer's theorem [17]. If a Gaussian kernel is used, (20) can be written as

$$f(x)=\text{sign}\left(\sum_{i\in\text{SV}}w_i'\exp\left(-\frac{\|\ \vec{x}-\vec{x}_i\ \|^2}{\sigma^2}\right)+b\right). \tag{21}$$

B. Parameter Learning by SVM

This section introduces how linear SVM is applied to learn consequent part parameters $\vec{a}$ in FS-FCSVM. Each input entry $\vec{x}$ is transformed to the firing strength feature vector $\vec{\mu}=[\mu_1,\mu_2,\ldots,\mu_r]$ in Layer 3. The firing strength $\vec{\mu}$ is fed as input for linear SVM learning, and the original training pairs in (5) are represented by

$$S=\left\{\left(\vec{\mu}\ (\vec{x}_1),y_1\right),\left(\vec{\mu}\ (\vec{x}_2),y_2\right),\ldots,\left(\vec{\mu}\ (\vec{x}_N),y_N\right)\right\}. \tag{22}$$

From (18) and (22), the following decision function can be obtained:

$$\begin{aligned}f(\vec{x})&=\text{sign}\left(y'(\vec{x})\right)\\&=\text{sign}\left(\sum_{i=1}^{N}y_i\alpha_i\left\langle\vec{\mu}\ (\vec{x}),\vec{\mu}\ (\vec{x}_i)\right\rangle+b\right)\end{aligned} \tag{23}$$

Fig. 2. Examples of training images used for training in experiments 1–3.

where α_i is solved by (16). After solving α_i, the above equation can be expressed as follows:

$$\begin{aligned}\text{sign}\left(y'(\vec{x})\right) &= \text{sign}\left(\sum_{i=1}^{N} y_i\alpha_i \sum_{k=1}^{r} \mu_k(\vec{x})\mu_k(\vec{x}_i) + b\right) \\ &= \text{sign}\left(\sum_{k=1}^{r}\left(\sum_{i=1}^{N} y_i\alpha_i\mu_k(\vec{x}_i)\right)\mu_k(\vec{x}) + b\right) \\ &= \text{sign}\left(\sum_{k=1}^{r} a_k\mu_k(\vec{x}) + b\right) \\ &= \text{sign}\left(\sum_{k=1}^{r} a_k \exp\left(-\frac{\| \vec{x} - \vec{m}_k \|^2}{\sigma_k^2}\right) + b\right).\end{aligned} \tag{24}$$

From the first line in (24), it can be found that numerous operations are required to compute the final output $y'(\vec{x})$. To reduce the operation numbers, coefficients of the variable $\mu_k(\vec{x})$ are combined from derivations in (24). The results in the last two lines of (24) show that the operation number is significantly reduced after rearranging the coefficients. The output value $y'(\vec{x})$ is the same as the output of FS-FCSVM in (8). That is, the consequent part parameter a_k in FS-FCSVM is solved, and

$$\begin{aligned}a_k &= \sum_{i=1}^{N} y_i\alpha_i\mu_k(\vec{x}_i) \\ &= \sum_{i\in \text{SV}} y_i\alpha_i\mu_k(\vec{x}_i).\end{aligned} \tag{25}$$

Equation (25) shows that despite the number of SVs in the linear SVM, only one single weight a_k is recorded for each rule after training. Thus, in (24), the memory required for storing consequent parameters a_k in FS-FCSVM depends only on the fuzzy rule number r. Note that the parameter C in (15) should be set *a priori* when a linear SVM is applied to FS-FCSVM consequent parameters learning. As in general SVM learning, the parameter C is heuristically chosen in this paper. That is, different values of C are tried, and the one with the best performance is chosen.

By comparing FS-FCSVM with Gaussian kernel SVM, it can be found that there are two distinguishing characteristics of FS-FCSVM, although both of them perform similar linear combination of Gaussian functions. First, the widths of all Gaussian kernels in Gaussian kernel SVM are the same [see (21)]; whereas in FS-FCSVM, the width σ_k varies for each Gaussian function [see (24)]. Second, (21) shows that the number of weights or Gaussian kernels in SVM is equal to the number of SVs; whereas in FS-FCSVM, it is equal to the number of fuzzy clusters. If the input variable dimension is n and there are N' SVs in the Gaussian kernel SVM, then the total number of parameters in it is $N'(n+1)+1$; whereas for FS-FCSVM with r rule number, the total number of parameters is $r(n+2)$. In general, the number of SVs N' is much larger than rule number r, as can be seen in the following image segmentation problem, so the number of parameters in FS-FCSVM is much smaller than SVM in application. This characteristic not only helps to reduce memory but also improves computation speed.

VI. Experimental Results

Two databases are used to test performance of the proposed FS-FCSVM segmentation method. The first database contains only yellow skin color, and the second database contains skin color of different races. Experimental results are presented in the following sections.

A. Yellow Skin Color Segmentation

Yellow skin color segmentation by FS-FCSVM is experimented in this section. Training data are collected from 50 images with the resolution of 640 × 480 pixels, and the percentage of all skin pixels (i.e., the global amount of skin pixels divided by global amount of the pixels in the image) is equal to 9.2031%. Six of the 50 training images are shown in Fig. 2. To verify the generalization ability of the proposed method, 500 images are used for testing, where each image consists of 640 × 480 pixels. That is, there are 1.536×10^8 pixels in total, which should be enough for generalization performance test. The percentage of skin pixels in these 500 test images is equal to 13.35%. Experiments are performed on a personal computer with an Intel 2.0-GHz central processing unit, Windows XP operating system, and the programs are written in Visual C++. For performance evaluation, the true

Fig. 3. Examples of testing images used in experiments 1–3.

skin color pixels in these test images are manually segmented. Twelve of the 500 test images are shown in Fig. 3.

To measure the segmentation performance, the receiver operating characteristic (ROC) curve is used. In the following ROC figures, the vertical axis is labeled as "detection rate (DR)," which gives the fraction of skin pixels being correctly classified, and the horizontal axis is labeled as "false positive rate (FPR)," which gives the fraction of nonskin pixels being mistakenly classified as skin. A larger area under the ROC curve means a better segmentation performance.

Two types of color features are used in the experiments. In experiment 1, the h and S values of each pixel are used. In experiment 2, the average h and S values over a 10×10 nonoverlapping block of pixels are used. In experiment 3, the average h and S values over a 10×10 block with the consideration of neighboring block pixels are examined. It was stated in [29] that variations in illumination chromaticity in test databases are important for testing system robustness. Experiments 4 and 5 test the performance sensitivity in real-life conditions, using the images suffering from systematic color appearance shifts (both lighter and darker).

For FS-FCSVM, structure learning coefficient $\mu_{\rm th}$ and parameter learning coefficient C in the SVM should be assigned in advance. In the following experiments, coefficient $\mu_{\rm th}$ is set to 0.75 according to a compromise between system size and performance. The best coefficient C is found to be 20 according to experiments on different C values.

Experiment 1: The h and S values of each pixel are directly used as features in this experiment. One hundred skin color pixels and 100 nonskin color pixels are randomly selected from each of the 50 training images (i.e., 10 000 pixels were collected for training). There are two inputs, i.e., h and S values, in the FS-FCSVM. When the input is skin color, then the desired output is "+1"; otherwise, the desired output is "−1." When an unknown pixel is fed to the input of FS-FCSVM, the output y is computed. If $y \geq \theta$, where θ is a predefined segmentation threshold, the corresponding pixel is classified as "skin color"; otherwise, it is classified as "nonskin color." The number of fuzzy rules after training is 30. The number of parameters in FS-FCSVM and classification rate over the 10 000 training

TABLE I
COMPARISON OF THE METHODS USED IN EXPERIMENT 1

method	SVM	FNN	MGC	FS-FCSVM
Structure	959 kernels	30 rules	30 Gaussians	30 rules
Parameter number	2878	150	150	120
Classification rate	97.06%	93.56%	88.14%	99.5%
Segmentation time	106 (sec)	3.15 (sec)	3.15 (sec)	3.08 (sec)

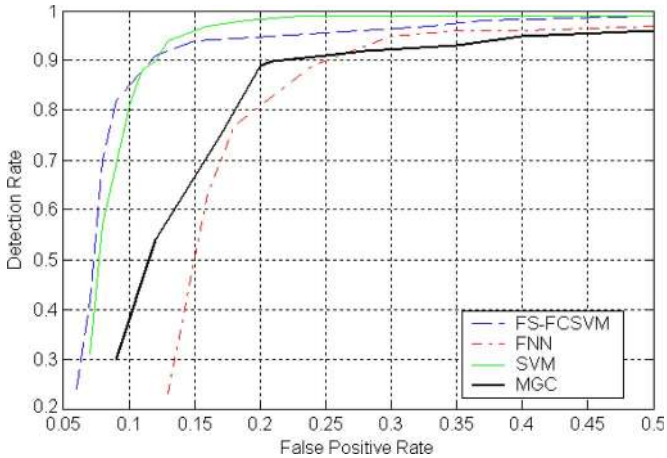

Fig. 4. ROC curves for test images with different segmentation methods using h and S values in experiment 1.

Fig. 5. Segmentation results for the test images in Fig. 3 by FS-FCSVM using h and S pixel values in experiment 1.

TABLE II
COMPARISON OF THE METHODS USED IN EXPERIMENTS 2 AND 3

Method	SVM	FNN	MGC	FS-FCSVM (Experiment 2)	FS-FCSVM (Experiment 3)
Structure	101 kernels	30 rules	30 Gaussians	30 rules	30 rules
Parameter number	304	150	150	120	120
Classification rate	96.3%	89.2%	86.3%	96.3%	96.3%
Segmentation time	0.116 (sec)	0.052 (sec)	0.052 (sec)	0.041 (sec)	0.043 (sec)

pixels, as well as the segmentation time for each 640 × 480 image by FS-FCSVM after the hS values of each pixel are obtained, are listed in Table I. The ROC curve for different values of θ for the 500 test images is shown in Fig. 4. Fig. 5 illustrates the segmentation results of the test images in Fig. 3. This figure shows that most of the misclassified pixels have colors similar to yellow skin color.

For comparison, Gaussian kernel SVM and FNN classifiers using the same feature are used in this experiment. For Gaussian kernel SVM, the best coefficient C is found to be two after several trials. The number of kernels, total number of parameters, training result, and segmentation time for one image are shown in Table I. The ROC curve is shown in Fig. 4. For FNN, the structure is also learned by the fuzzy clustering used in FS-FCSVM, and there are 30 rules in FNN. The parameters in antecedent and consequent parts of FNN are learned by gradient descent. The total number of parameters in FNN, training result, and segmentation time are shown in Table I. The ROC curve for test images is shown in Fig. 4. The results in Fig. 4 show that the performance of FS-FCSVM is comparable to Gaussian kernel SVM. However, FS-FCSVM shows the advantages of using much smaller number of parameters and faster segmentation time than SVM. The training result of FNN is slightly poorer than that of FS-FCSVM, when the same number of rules is used. However, Fig. 4 shows that the test result of FS-FCSVM is much better than FNN. The reason is that parameters in FNN are trained by minimizing empirical risk; whereas parameters in FS-FCSVM are trained by minimizing structural risk.

Another commonly used classifier, i.e., the MGC [6], [7], using hS pixel values, is also used for experiment. For MGC, a mixture density function is expressed as the sum of Gaussian kernels as follows:

$$P(\vec{x}) = \sum_{i=1}^{N} w_i \frac{1}{(2\pi)^{\frac{3}{2}} |\sum_i|^{\frac{1}{2}}} \times \exp\left(\frac{-(\vec{x} - \vec{\mu}_i)^T \sum_i^{-1} (\vec{x} - \vec{\mu}_i)}{2}\right) \quad (26)$$

where $\vec{x}$ is the hS color vector, and the contribution of the ith Gaussian is determined by a scalar weight w_i, mean vector $\vec{\mu}_i$, and diagonal covariance matrix $\sum_i$. We trained two separate mixture models for the skin and nonskin classes. For each model, 15 Gaussians are used so that there is a total of 30 Gaussians in MGC, with a Gaussian corresponding to a rule in FS-FCSVM and FNN. The models were trained using parallel implementation of the standard EM algorithm [7], [12]. We predefined a segmentation threshold θ, so that a particular mixture density function is labeled skin color if

$$\frac{\max P_{\text{skin}}(x)}{\max P_{\text{nonskin}}(x)} \geq \theta. \quad (27)$$

The number of parameters in MGC, training result, and segmentation time are shown in Table I. The ROC curve is shown in Fig. 4. The results show that the performance of MGC is much worse than FS-FCSVM. In fact, both the training and generalization performance using the statistical EM algorithm are not good when only 10 000 pixels are used for training. To achieve a better generalization result using EM algorithm, a much larger size of training data is required.

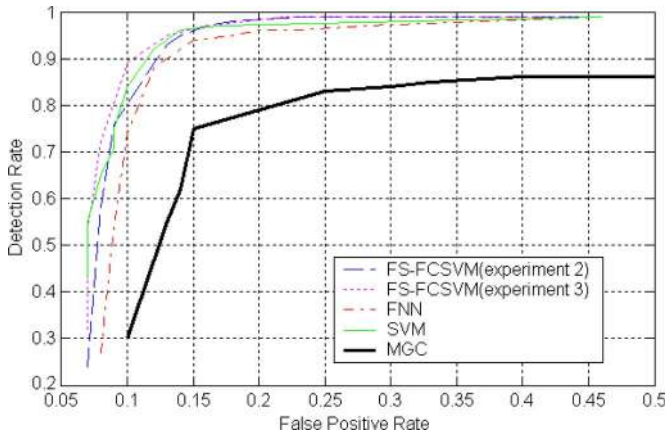

Fig. 6. ROC curves for test images with different segmentation methods using average hS values in a block in experiments 2 and 3.

Experiment 2: The average h and S values over 10 × 10 nonoverlapping blocks of pixels are used as features in this experiment. For training data collection, ten skin color blocks and ten nonskin blocks are randomly selected from each of the 50 training images, and there is a total of 1000 blocks for training. The average hS values of the 100 pixels in a block are used as features, and there are 1000 sets of average hS values used for training. The training results of FS-FCSVM are shown in Table II. For test image segmentation, scans of the whole image using nonoverlapping blocks are performed. When a block is classified as skin color, all pixels in such a 10 × 10 block are classified as skin color. The ROC curve for the test images is shown in Fig. 6, and the result is similar to that in experiment 1. The advantage of using average hS values of each block is that segmentation is performed on

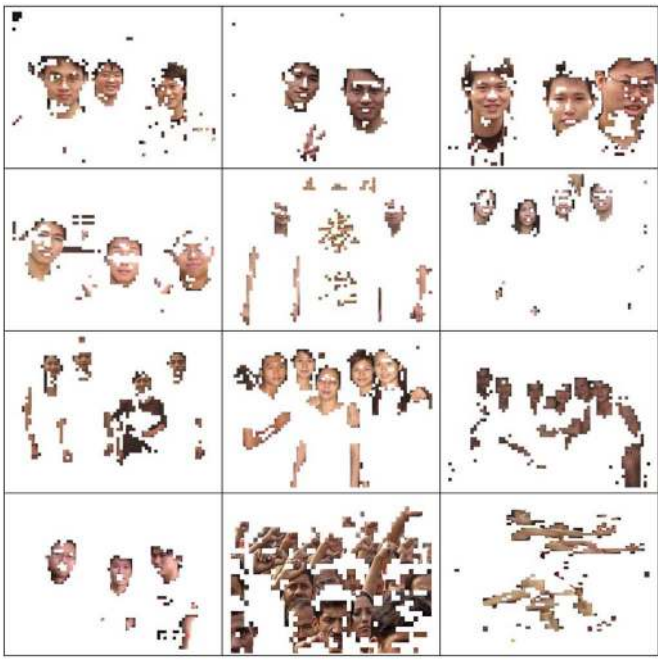

Fig. 7. Segmentation results for the test images in Fig. 3 by FS-FCSVM with average h and S values over a 10 × 10 nonoverlapping block of pixels in experiment 2.

blocks instead of pixels, so the segmentation time is reduced. In this experiment, classification is performed 3072 times for each image in contract to 307 200 times in experiment 1. The segmentation time for one image in this experiment is shown in Table II. The result shows that segmentation performed on blocks is much faster than on pixels. This time consideration is particularly important when real-time classification is performed. Fig. 7 illustrates the segmentation results obtained using the test images in Fig. 3. There are fewer noise in Fig. 7 when compared with the result in Fig. 5, and the cost is a lower segmentation resolution in Fig. 7.

Segmentation by Gaussian kernel SVM, FNN, and MGC using the same average h and S values of each block are also experimented. Training results and segmentation time are shown in Table II. ROC curves of these methods are shown in Fig. 6. As in experiment 1, the performance of SVM is similar to that of FS-FCSVM at the cost of using a much lager number of parameters and slower segmentation time. Performance of FNN and MGC are worse than FS-FCSVM when similar numbers of parameters are used in these classifiers.

Experiment 3: The average h and S values over a 10 × 10 block of pixels are also used as features in this experiment except that each block is classified according to the average of the classification results of four neighboring blocks. The trained FS-FCSVM in experiment 2 is used as the classifier. For test image segmentation, scans of the whole image using nonoverlapping blocks are performed. The outputs of FS-FCSVM of these nonoverlapping blocks are recorded. The next classification approach is illustrated in Fig. 8. To classify block A, the four FS-FCSVM classification outputs of neighboring blocks B1, B2, B3, and B4 are computed and averaged. If this average value is larger or equal to θ, then all of the pixels in block A are classified as skin color. Except the

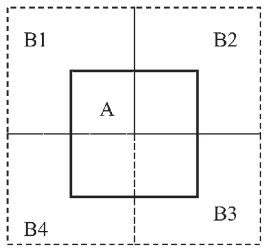

Fig. 8. Classification of block A is based on the average of classification results of blocks B1–B4.

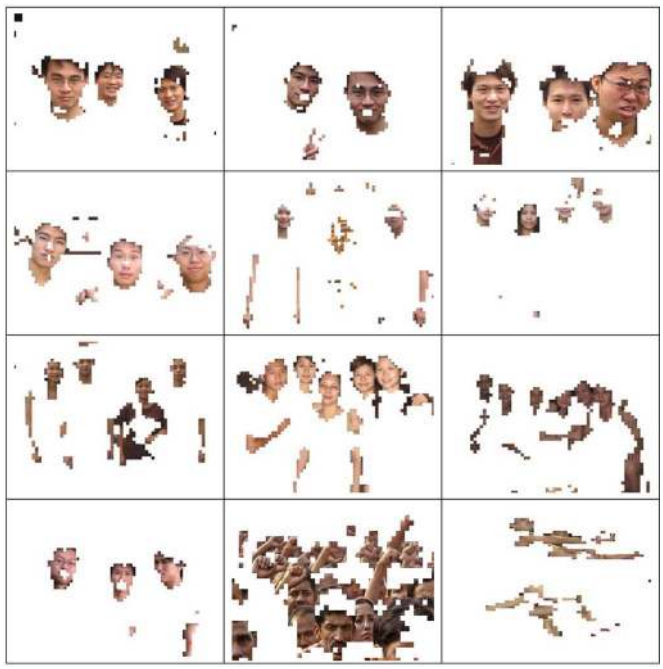

Fig. 9. Segmentation results for the test images in Fig. 3 by FS-FCSVM in experiment 3.

boundary blocks, where the FS-FCSVM output is directly used for classification, all of the nonoverlapping blocks are classified in the same way. The ROC curve is shown in Fig. 6. Compared with experiment 2, the results show that about 10% DR rate is improved when FPR is around 0.1. For statistically significant test, the t-test on the 500 test images is used. Denote the average DR $(\mathrm{FPR} = 0.1)$ of FS-FCSVM with and without neighborhood averaging by m_{ave} and m_{non}, respectively. The standard deviations of FS-FCSVM with and without neighborhood averaging are 7.3% and 8.8%, respectively. With significance level $\alpha = 1\%$, the hypothesis $m_{\mathrm{non}} > m_{\mathrm{ave}}$ against the alternative $m_{\mathrm{non}} = m_{\mathrm{ave}}$ is accepted. Table II shows that the improvement is at the cost of only a slightly larger computation time. Fig. 9 illustrates the segmentation results of the test images in Fig. 3. There are fewer noises in Fig. 9 when compared with the segmentation results in Fig. 7. That is, the averaging operation helps to eliminate noise. In addition, the segmented skin regions in Fig. 9 are more compact than those in Fig. 7, which is helpful in applications such as face detection and gesture recognition.

Experiment 4: Experiments on images with varying lighting conditions are performed to test robustness of the proposed FS-FCSVM method. For fair comparison with performance

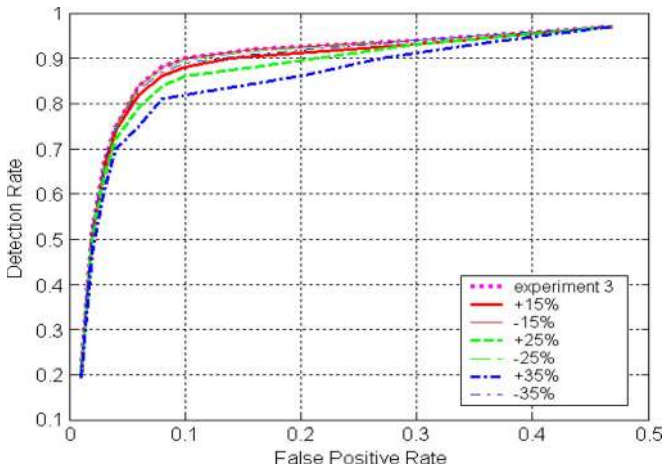

Fig. 10. ROC curves for the test database in experiment 3, its lighter (+15%, +25%, and +35%) and darker (−15%, −25%, and −35%) versions using FS-FCSVM with neighborhood averaging in experiment 4.

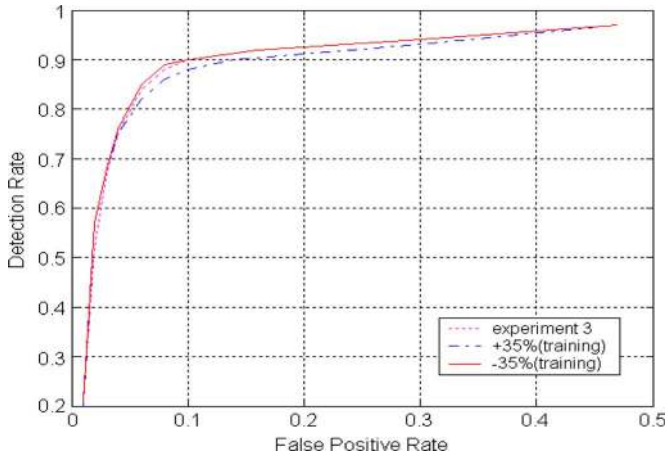

Fig. 11. ROC curves of FS-FCSVM (with neighborhood averaging) trained by the 1000 sets of training pixels in experiment 3, the lighter version (+35%) of the training pixels and the darker version (−35%).

using the original test database, new test databases are created by modifying the original test database with color appearance shifts (both lighter and darker). The preliminary RGB values in original test image database are all increased and decreased by a certain percentage to create lighter and darker image databases, respectively. This color shifting method was used in [26] for lighting compensation in contrast to lighting robustness test in this paper. Three new lighter image databases are created by increasing the original RGB values by 15%, 25%, and 35%. Note that when the RGB values are all larger than 255, a saturation white color is obtained. Another three new darker image databases are created by decreasing the original RGB values by 15%, 25%, and 35%. The same trained FS-FCSVM in experiment 3 is applied to these six new databases, and segmentation process is the same as that used in experiment 3. ROC curves of these six new databases and the original one tested in experiment 3 are shown in Fig. 10 for comparison. The ROC curves in Fig. 10 show that the segmentation method is robust to darker images and is more sensitive to lighter images. One major reason is that for some outdoors images, increment in lightness of skin pixels may generate near-white color pixels and result in wrong detection. The phenomenon becomes more obvious when increment in lightness is larger.

Experiment 5: This experiment demonstrates how different training conditions affect the performance of the proposed method. A lighter version of the 1000 sets of training pixels used in experiment 2 is created by increasing the RGB values of each pixel by 35%. Then, average hS values of these new 1000 sets of training pixels are used for FS-FCSVM training. Training process of FS-FCSVM is the same as that in experiment 2. The segmentation process and test database in experiment 3 are also used to measure the trained FS-FCSVM performance, and the corresponding ROC curve is shown in Fig. 11. Similarly, training of FS-FCSVM using a darker version of the 1000 sets of training pixels used in experiment 2 is also performed. The ROC curve of this newly trained FS-FCSVM when applied to the same test database in experiment 3 is also shown in Fig. 11. The results in Fig. 11 show that performance using the original training data and its darker version is very similar. The performance degrades when a lighter training database is used. The explanation for this result is that lightness increment, in some of the training pixels, may generate saturation white color pixels as in experiment 4. Training of these white pixels as skin colors degrades the test performance.

Fig. 12. Twelve of the 500 images for testing in experiment 6.

B. Skin Color Segmentation of People of Different Races

The database used is the California Institute of Technology face database [30]. There are 450 images in total, which contains 27 different people of different races and facial expressions. Each image consists of 896 × 592 pixels. As in Section VI-A, training data are also collected from 50 images, where ten skin color blocks and ten nonskin blocks are randomly selected from each of the 50 training images, and there is a total of 1000 blocks for training. The left 400 images are used for testing. Twelve of the 500 test images are shown in Fig. 12. The average h and S values over a 10 × 10 nonoverlapping block of pixels are used as features, as in experiment 2. Performance of FS-FCSVM, Gaussian kernel SVM, FNN, and MGC are presented in the following experiment.

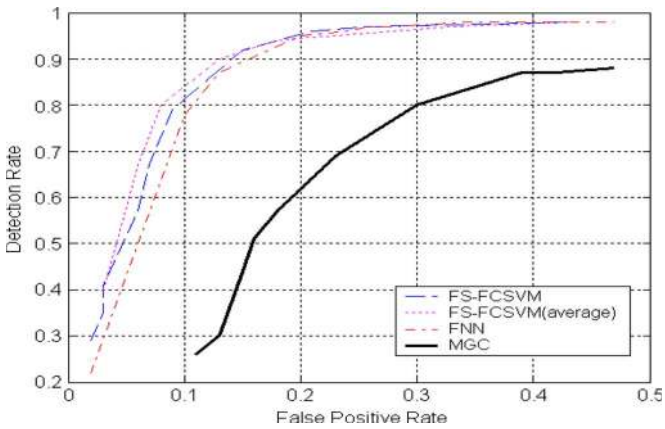

Fig. 13. ROC curves of different methods in experiment 6.

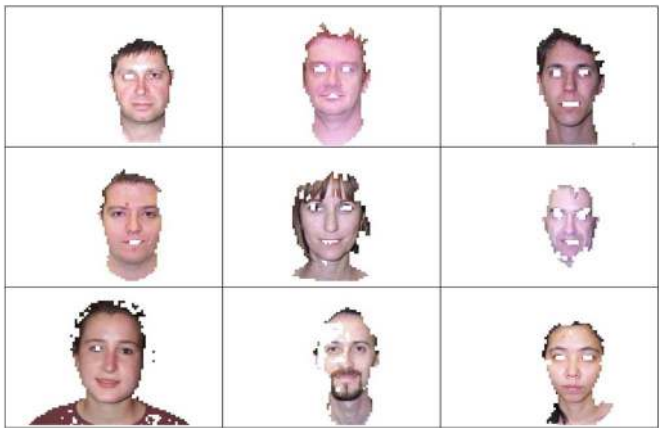

Fig. 14. Segmentation results for the test images in Fig. 13 by FS-FCSVM with neighborhood averaging in experiment 6.

Experiment 6: Parameters for FS-FCSVM training are the same as those used in Section VI-A. After training, there are 30 rules in FS-FCSVM. For FS-FCSVM, segmentations of an image with and without neighborhood averaging (as in experiments 2 and 3) are performed. The ROC curves are shown in Fig. 13. Fig. 14 illustrates the segmentation results of the test images in Fig. 12 by FS-FCSVM with neighborhood averaging. For comparison methods, there are 30 rules in FNN and 30 Gaussians in MGC. ROC curves of these comparison methods are shown in Fig. 13. As in experiment 2, FS-FCSVM shows better performance than FNN and MGC. As in experiment 3, FS-FCSVM with neighborhood averaging shows a better performance than that without neighborhood averaging. Since segmentation on skin colors of multiple races is performed, Fig. 13 shows that the performance of FS-FCSVM with neighborhood averaging is a little worse than that in experiment 3 where segmentation on yellow skin color is performed.

VII. Conclusion

This paper proposes skin color image segmentation by FS-FCSVM. Compared with the two widely used FNN and SVM classifiers, we have shown the advantages of FS-FCSVM. With a similar number of parameters in FS-FCSVM and FNN, the test result shows that the FS-FCSVM achieves higher generalization ability. With similar segmentation performance, both the number of parameters in FS-FCSVM and computation time are much smaller than in Gaussian kernel SVM. In other words, structure learning by fuzzy clustering helps to reduce the number of rules in FS-FCSVM. Parameter learning by linear SVM is based on structural risk minimization (or minimizing the test error) and helps to keep high generalization ability in FS-FCSVM. Performance of FS-FCSVM using different types of features based on hS color model is tested. In the future, the use of other color models and lighting compensation techniques will be studied to further improve segmentation performance. In application, we will also apply the proposed method to gesture recognition and face detection problems.

References

[1] W. Skarbek and A. Koschan, "Colour image segmentation—A survey," Tech. Univ. Berlin, Berlin, Germany, Tech. Rep., Oct. 1994.
[2] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognit.*, vol. 34, no. 12, pp. 2259–2281, Sep. 2000.
[3] M. J. Swain and D. H. Ballard, "Indexing via color histograms," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, Nov. 1991.
[4] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *Int. J. Comput. Vis.*, vol. 46, no. 1, pp. 81–96, Jan. 2002.
[5] J. Han and K. K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Trans. Image Process.*, vol. 11, no. 8, pp. 944–952, Aug. 2002.
[6] S. Mckenna, S. Gong, and Y. Raja, "Modeling facial colour and identity with Gaussian mixtures," *Pattern Recognit.*, vol. 31, no. 12, pp. 1883–1892, Dec. 1998.
[7] M. H. Yang and N. Ahuja, "Gaussian mixture model for human skin color and its application in images and video databases," in *Proc. SPIE—Conf. Storage Retrieval Image Video Databases*, 1999, vol. 3656, pp. 458–466.
[8] E. Littmann and H. Ritter, "Adaptive color segmentation—A comparison of neural and statistical methods," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 175–185, Jan. 1997.
[9] P. Campadelli, D. Medici, and R. Schettini, "Color image segmentation using Hopfield networks," *Image Vis. Comput.*, vol. 15, no. 3, pp. 161–166, Mar. 1997.
[10] Y. S. Lo and S. C. Pei, "Color image segmentation using local histogram and self-organization of Kohonen feature map," in *Proc. Int. Conf. Image Process.*, Oct. 1999, vol. 3, pp. 232–235.
[11] R. Rae and H. Ritter, "Recognition of human head orientation based on artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 2, pp. 257–265, Mar. 1998.
[12] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, Apr. 1984.
[13] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, May 1996.
[14] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 12–32, Feb. 1998.
[15] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[16] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[17] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[18] B. Schölkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2758–2765, Nov. 1997.
[19] J. H. Chiang and P. Y. Hao, "Support vector learning mechanism for fuzzy rule-based modeling: A new approach," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 1, pp. 1–12, Feb. 2004.
[20] Y. Chen and J. Z. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 716–728, Dec. 2003.
[21] R. Lukac, B. Smolka, K. Martin, K. N. Plataniotis, and A. N. Venetsanopoulos, "Vector filtering for color imaging," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 74–86, Jan. 2005.

[22] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon—Int. Conf. Computer Graphics Vision*, Moscow, Russia, Sep. 2003, pp. 85–92.
[23] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen, "Adaptive skin color modeling using the skin locus for selecting training pixels," *Pattern Recognit.*, vol. 36, no. 3, pp. 681–690, Mar. 2003.
[24] J. C. Terrillon, A. Pilpre, Y. Niwa, and K. Yamamoto, "Analysis of a large set of color spaces for skin pixel detection in color images," in *Proc. SPIE—6th Int. Conf. Quality Control Artificial Vision*, May 2003, vol. 5132, pp. 433–446.
[25] 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," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recog.*, Mar. 2000, pp. 54–61.
[26] R. L. Jsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 696–706, May 2002.
[27] H. Stern and B. Efros, "Adaptive color space switching for face tracking in multi-colored lighting environments," in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recog.*, May 2002, pp. 236–241.
[28] L. Sigal, S. Sclaroff, and V. Athitsos, "Skin color-based video segmentation under time-varying illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 7, pp. 862–877, Jul. 2004.
[29] B. Martinkauppi, M. Soriano, and M. Pietikäinen, "Comparison of skin color detection and tracking methods under varying illumination," *J. Electron. Imaging*, vol. 14, no. 4, pp. 043 014-1–043 014-19, Oct. 2005.
[30] *CIT Face Database*. [Online]. Available: http://www.vision.caltech.edu/Image_Datasets/faces/

Chia-Feng Juang (M'00) received the B.S. and Ph.D. degrees in control engineering from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1993 and 1997, respectively.

In 2001, he joined the Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, where he is currently a Professor of electrical engineering. His current research interests include computational intelligence, intelligent control, computer vision, speech signal processing, and FPGA chip design.

Dr. Juang is the recipient of the Youth Automatic Control Engineering Award from the Chinese Automatic Control Society, Taiwan, in 2006.

Shih-Hsuan Chiu received the B.S. degree in electrical engineering from the National Chung Hsing University, Taichung, Taiwan, R.O.C., in 2005.

In 2005, he joined the Realtek Semiconductor Corporation, Hsinchu, Taiwan, where he is currently an R&D Engineer. His current research interests include image processing and support vector machines.

Shen-Jie Shiu received the B.S. degree in electrical engineering from the National Chung Hsing University, Taichung, Taiwan, R.O.C., in 2006.

He is currently in military service in Taiwan. He is also with the Department of Electrical Engineering, National Chung Hsing University. His research interests include computer vision and support vector machines.