Journal of Electronic Imaging 15(1), 013015 (Jan–Mar 2006)

# Efficient color face detection algorithm under different lighting conditions

**Tze-Yin Chow**
**Kin-Man Lam**
**Kwok-Wai Wong**
The Hong Kong Polytechnic University
Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
Hong Kong, China

**Abstract.** *We present an efficient and reliable algorithm to detect human faces in an image under different lighting conditions. In our algorithm, skin-colored pixels are identified using a region-based approach, which can provide more reliable skin color segmentation under various lighting conditions. In addition, to compensate for extreme lighting conditions, a color compensation scheme is proposed, and the distributions of the skin-color components under various illuminations are modeled by means of the maximum-likelihood method. With the skin-color regions detected, a ratio method is proposed to determine the possible positions of the eyes in the image. Two eye candidates form a possible face region, which is then verified as a face or not by means of a two-stage procedure with an eigenmask. Finally, the face boundary region of a face candidate is further verified by a probabilistic approach to reduce the chance of false alarms. Experimental results based on the HHI MPEG-7 face database, the AR face database, and the CMU pose, illumination, and expression (PIE) database show that this face detection algorithm is efficient and reliable under different lighting conditions and facial expressions. © 2006 SPIE and IS&T.* [DOI: 10.1117/1.2179080]

## 1 Introduction

Face detection is the first step in any face processing system. Yang, Kriegman, and Ahuja[1] gave a survey of the current face detection technology and provided a definition of face detection. This is a challenging task, because the human face is a highly variable 3-D object with different perspective and uneven illumination. The detection performance may be affected by the presence of glasses, different races, genders, facial hair, facial expressions, lighting conditions, etc. As a result, a true face may not be well detected.

Numerous approaches have been proposed to detect human faces in gray-level images. These approaches usually search over a range of scales and locations for possible human faces and verify the patterns with a pattern classifier. Moghaddam and Pentland[2] applied the eigenface decomposition technique to reduce face image from a high-dimensional image space to a lower dimensional; the images are then trained with the expectation-maximization algorithm to optimize the mixture parameters. Multiscale saliency maps based on maximum likelihood are then computed for single face detection. Liu[3] purposed using Bayesian discriminating features to detect multiple faces in an image. Face and nonface classes are trained to discriminate input images during detection. 4500 nonface class patterns, which lie close to the face class, are generated from nine natural images that do not contain any human faces. Papageorgiou and Poggio[4] proposed an example-based learning approach by using a large set of positive and negative examples to implicitly derive an object model class to be classified with the support vector machine. Mohan, Papageorgiou, and Poggio[5] presented an example-based framework for detecting objects in static images by components in two steps. The example-based component detectors are first trained to find components separately, including the human body, head, legs, left arm, and right arm. Then, a second example-based classifier combines the results of the component detectors to classify a pattern as either a "person" or a "nonperson." Rowley, Baluja, and Kanade[6] developed a neural network-based face detection system, which examines an image via small windows and decides whether the windows contain a face or not.

Recently, face segmentation based on the color-based approach has received mass attention, since color provides more information than grayscale intensity, especially when the skin is under different illumination conditions. The basic idea is that skin colors for people of different races and illumination are distributed more or less in the same region in a color space. Consequently, the search for faces can be restricted within the skin-color regions. Chai and Ngai[7] proposed a skin-color segmentation method under a range of luminance and chrominance. Hsu, Abdel-Mottaleb, and Jain[8] proposed a face detection algorithm for color images under varying lighting conditions based on a lighting compensation technique and a nonlinear color transformation to detect skin regions and to generate face candidates based on a spatial arrangement of the skin patches. Eyes, mouth, and boundary maps based on luminance and chrominance are constructed to verify a face candidate. Wong, Lam, and Siu[9] proposed to perform skin-color segmentation according to different illumination conditions, using the genetic algorithm[10] to search for possible face candidates. Greenspan, Goldberger, and Eshet[11] presented a mixture-of-Gaussians distribution to model the color distribution of shadowed face images.

Paper 05057R received Apr. 8, 2005; revised manuscript received Jun. 20, 2005; accepted for publication Jul. 26, 2005; published online Mar. 9, 2006.
1017-9909/2006/15(1)/013015/10/$22.00 © 2006 SPIE and IS&T.

In this work, we found that the red component of skin color becomes saturated under strong illumination. Therefore, to further improve the performance of skin-color detection, a color compensation scheme[12] is proposed to extend the range of red component to its saturated level. Then, the skin color is modeled with the mixture-of-Gaussians model to segment skin-color regions according to different illumination conditions. The performance of our algorithm is evaluated with face images with varying lighting conditions, different scales, facial expression, and with glasses. Experimental results show that our proposed algorithm is very fast and can achieve a high detection rate. The details of our approach for face detection is described in the following sections.

## 2 Human Face Detection

Our approach to detecting face regions in a color image consist of three steps. The first step is to segment the face color using the mean-shift algorithm.[13] The segmented regions are then processed by a color compensation scheme, and the skin-color distributions under different illuminations are modeled by means of the maximum-likelihood method. In the second step, our algorithm focuses on searching possible eye candidates within the segmented skin-color regions. Possible face candidates are formed by grouping pairs of eye candidates. Finally, a two-step procedure based on an eigenmask for face verification is performed. Once the face has been verified and short listed, the face contour is further verified with a Gaussian function to further improve the reliability and accuracy of our face detection algorithm.

### 2.1 *Face-Color Segmentation*

Color information has been a commonly used technique for segmenting human face regions from a complex background. Skin-like regions are extracted using both the normalized RGB color model and the hue, saturation, value (HSV) color model.[14] The chrominance information in the YCbCr color space is used for the segmentation of skin-like regions.[7] However, these methods can achieve good results if the face images are captured under good lighting conditions. We therefore propose a robust color compensation method[12] with the use of the mixture-of-Gaussian model[11] to represent skin color under various illuminations.

#### 2.1.1 *Color image segmentation with mean-shift algorithm*

The mean-shift algorithm[13] is a kernel-based density estimation technique that has been used in many applications including data clustering, image segmentation, object tracking,[15] etc. In particular, it is a nonparametric and robust technique to analyze feature spaces.

In color image segmentation, the feature space can be considered as an empirical probability density function (PDF). Dense regions in the feature space are considered as local maxima of the PDF, that is, the modes of the unknown density. Once the location of a mode is determined, the cluster associated with the mode is delineated based on the local structure of the feature space.

Given $n$ training color vectors $x_i$, $i=1,\ldots,n$, in the $d$-dimensional space $\Re^d$. Each vector contains the $x$ and $y$ coordinates and the three color components of the skin-color pixel concerned, so dimension $d$ is 5. The feature space can then be modeled by an unknown kernel density function $K$ as follows:

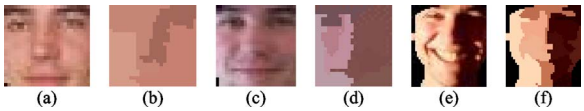


**Fig. 1** Face images under various lighting conditions: (a) normal lighting, (b) segmented faces in (a), (c) dark and side light, (d) segmented faces in (c), (e) strong overhead light, and (f) segmented faces in (e).

$$K_{h_s h_r}(x) = \frac{C}{h_s^2 h_r^3} \sum_{i=1}^{n} K\left(\left\| \frac{x^s - x_i}{h_s} \right\|\right) K\left(\left\| \frac{x^r - x_i}{h_r} \right\|\right), \tag{1}$$

where $x^s$ is the spatial part, $x^r$ is the color part of a feature vector, $k(x)$ is the common profile in both domains, $h_s$ and $h_r$ are the corresponding kernel bandwidths, and $C$ is a constant for normalization. The sample mean at $x$ is defined as:

$$m(x) = \frac{\sum_{i=1}^{n} x_i K\left(\left\| \frac{x - x_i}{h} \right\|\right)}{\sum_{i=1}^{n} K\left(\left\| \frac{x - x_i}{h} \right\|\right)}. \tag{2}$$

The vector $x$ is updated in the form of iteration such that $x \leftarrow m(x)$ with $m(x)=\{m(x); x \in \Re^d\}$. The difference $m(x) - x$ is called the mean shift, and the repeated movement of data points to the sample mean is called the mean-shift algorithm. The mean-shift algorithm iterates until it is converged with zero gradient, i.e., $m(x)-x=0$. The convergence is guaranteed at a nearby point. Once the mean-shift algorithm is converged, the local mean is shifted toward the region where the majority of the points reside, that is, the local maxima or the mode of the region. Our color image segmentation algorithm consists of two steps: 1. mean-shift filtering, which smoothes the input image by running the mean-shift algorithm; and 2. mean-shift segmentation, which delineates the smoothed image and purges the small regions.

#### 2.1.2 *Skin-color modeling*

A skin-color model can be trained by learning from a large set of sample skin-color pixels in the preprocessing stage. For face color under varying illumination, the skin-color distribution is no longer unimodal. In fact, the face color is distributed from shadowed faces to strong overhead light-projected faces, as well as faces under normal lighting conditions. One way to tackle this varying distribution is to use a Gaussian mixture model.

In our approach, the training face images are extracted from the HHI MPEG-7 face database, and each face image is segmented with the mean-shift algorithm, as described in the previous section. After segmenting a face image into a number of regions, the modes of the skin-color pixels are left in the respective facial regions. This can ensure that there is no outlier skin color in the training set. Figure 1

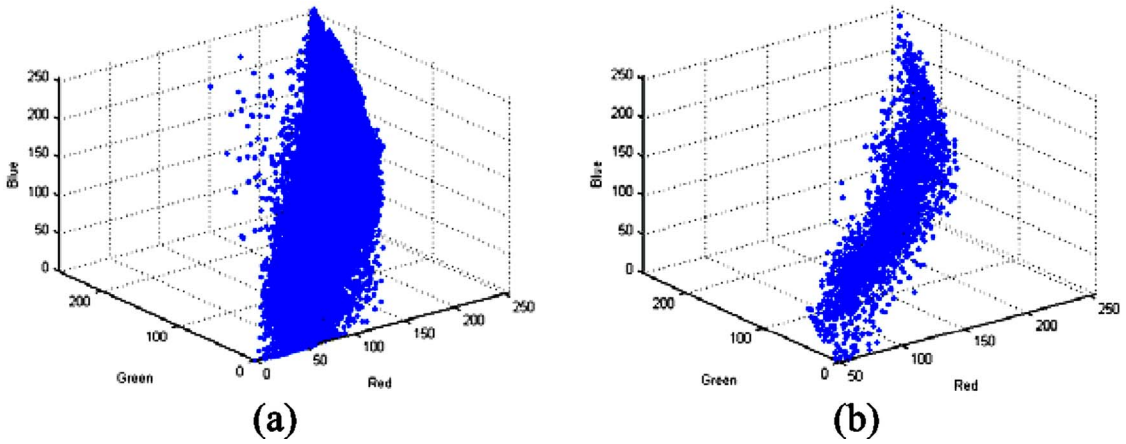


**Fig. 2** RGB distributions of the face color: (a) RGB distribution of original faces, and (b) RGB distribution of segmented faces.

shows some face images and their corresponding segmented results under various lighting conditions using the mean-shift algorithm.[13]

A total of 366,431 skin-color pixels were extracted from 206 faces in the HHI MPEG-7 face database. After color image segmentation with the mean-shift algorithm, only 2425 skin-color pixels were left. The average number of regions in each segmented face is 11.7, i.e., 11.7 pixels per segmented face on average. Figures 2(a) and 2(b) show the color distributions of the original faces and segmented faces. In Fig. 2(a), there are pixels other than the skin-color distribution. Obviously, this is due to colors other than the skin color, such as eyeball, eyebrow, mustache, lip, etc. In Fig. 2(b), there are only a few pixels apart from the skin-color distribution. After segmentation, the eyes, mouth, and eyebrows are removed and segmented to become skin-like colors, as shown in Fig. 1. By observing the change of skin-color distribution in Fig. 2(a), the mean-shift process sharpens the skin-color pixel distribution in Fig. 2(b) to converge toward the mean of skin colors. This leads the skin colors in uneven lighting to converge to an optimal pixel value, thus less variance will appear under extreme illumination conditions.

*K-means face-color clustering*. Before applying the Gaussian mixture model to describe the distributions of the segmented skin colors in Fig. 2(b), we have to cluster the skin colors to obtain the initial parameters and the number of clusters for the Gaussian mixture model. Each of the clusters should clearly represent the face color under different lighting conditions. The *K*-means algorithm is applied to cluster the skin-color pixels for $k \geqslant 1$. As the RGB color space does not contain any information about lighting, the face color is first translated to the compensated YCbCr color space.[12] The face color is then divided into $k+1$ decision regions; $k$ face-color regions and the complementary nonface region. In our approach, we set $k=5$, which can empirically segment the skin and nonskin color well throughout our experiments.

*Face-color modeling via Gaussian mixture model*. With the results from the *K*-means algorithm, the skin-color distribution can be modeled using the Gaussian mixture model, which is a semiparametric model. To optimize the mixture parameters, maximum likelihood can be used to seek the best parameters based on the results from the *K*-means algorithm. The expectation-maximization (EM) algorithm[11,16] is used to estimate the many-to-one mapping from the Gaussian mixture model.

Based on the skin-color model, a pixel in a color image is determined to be of skin color or not by the following steps.

- Each pixel in the RGB color format is first translated to the compensated YCbCr color space.
- With the input color vector $x=(y,cb,cr)$, the Gaussian PDFs $f_j$, $j=1,\ldots,k$ are computed to find the one with the highest probability, i.e.,

$$\arg\max_j f(x|\alpha_j,\mu_j,\Sigma_j) = \alpha_j \frac{1}{[(2\pi)^d|\Sigma_j|]^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_j)^{\mathrm{T}}\Sigma_j^{-1}(x-\mu_j)\right\},$$

  where $\alpha_j$ is the probability, $\mu_j$ is the mean, and $\Sigma_j$ is the covariance matrix.
- Suppose that $f(x|\alpha_j,\mu_j,\Sigma_j)$ results in the highest probability. The following discriminant function is computed to obtain the face mask $P_{\mathrm{face}}(x)$,

$$P_{\mathrm{face}}(x) = \begin{cases} 1 & \text{if } [(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j) < \text{threshold}] \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

  where 1 represents face color and 0 is nonface color.

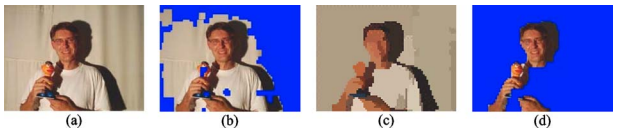


**Fig. 3** Skin-color segmentation under uneven lighting conditions: (a) original image, (b) skin-color segmentation with pixel-by-pixel approach, (c) color image segmentation with mean-shift algorithm, and (d) skin-color segmentation with region-based approach.

For a probability that is less than a predefined threshold, the color vector $x$ will be declared a nonskin color; otherwise, the color vector $x$ is skin color. The distribution of $x$ is modeled by a Gaussian function. A suitable threshold can therefore be determined empirically, and is set equal to one standard deviation of each Gaussian distribution in our experiment. With this threshold, the skin-color region and background can be segmented reliably.

*Region-based skin-color segmentation.* Traditional skin-color segmentation is performed based on a pixel-by-pixel approach. Each pixel in an image is checked to determine whether it is of skin color or not. After this skin-color segmentation process, some small holes will be introduced in the eye, nose, and mouth regions. These small holes can be removed using the morphological operations. However, some big holes in the background cannot be filled, as shown in Fig. 3(b). The result will worsen when the image concerned has varied lighting conditions.

Because of the drawback in the pixel-by-pixel approach, a region approach is proposed for skin-color segmentation. In our algorithm, an image is scaled to $80\times60$ to reduce the computational complexity. Other reduced resolutions can also be used, and this should be dependent on the resolution of the original image and the range of the face sizes to be detected. After the reduction, the reduced face size should be at least equal to $8\times8$. The color image is then segmented by the mean-shift algorithm into many regions, which are represented by the mode of the color pixels, as shown in Fig. 3(c). The details of the eyes, nose, and mouth are merged with the skin color. Maximum likelihood by means of a mixture of Gaussians with the optimized mixture parameters obtained by the EM algorithm is then applied. If the probability of a mode is larger than a certain threshold, the corresponding color will be classified as a skin color, and thus, the whole region will be declared a skin-color region. Compared to the pixel-by-pixel method, the region-based approach can achieve a better performance level under uneven lighting conditions, as shown in Fig. 3(d).

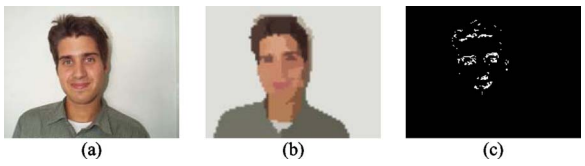


**Fig. 4** Possible eye candidate detection: (a) original image, (b) color image segmentation with mean-shift algorithm, and (c) possible eye candidate.

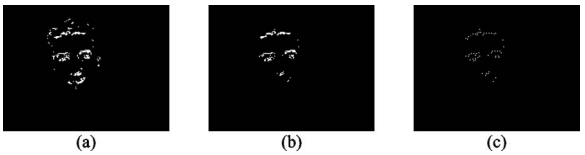


**Fig. 5** Possible eye candidate reduction: (a) possible eye candidates, (b) reduction of possible eye candidates by Sobel and red color distance, and (c) possible eye candidates in (b) grouped by $3\times3$ windows.

## 2.2 *Possible Eye Candidate Detection*

An efficient way to detect an eye region is by means of valley detection,[17] as the gray-level intensities in an eye region are relatively lower than in its neighborhood. However, valley detection fails when the eye region is shadowed under poor lighting conditions. In this section, we present an improved method that can detect eye regions more reliably under various lighting conditions.

Since the eyes are surrounded by skin, there is a significant color difference between the eye region and the skin color. Under the YCbCr color space, we can observe that the iris has a lower gray-level intensity, a higher Cb value, and a lower Cr value than the surrounding skin color. These kinds of properties will be used to determine whether a pixel in a segmented skin-color region is a possible eye candidate or not.

In our approach, an image is first segmented with the mean-shift algorithm at resolution $80\times60$, denoted as $I_{MS}(c)$, where $c=(y,cb,cr)$. Figure 4(b) shows the mean-shift segmented image with the eyes under overhead lighting. In the segmented image, the details of the eye are removed and replaced by skin-like color. Throughout our experiment, the original image $I(c)$ and the segmented image $I_{MS}(c)$ are scaled to $200\times150$. At this resolution, the computational complexity is reduced and the details of the eye regions are retained.

Suppose that the Y, Cb, and Cr components of a pixel in an image are denoted as $I(y)$, $I(cb)$, and $I(cr)$, respectively, while $I_{MS}(y)$, $I_{MS}(cb)$, and $I_{MS}(cr)$ denote the corresponding mean-shifted or segmented image. Based on the observed difference in color between the iris and the skin, $I(y)$ should be less than a certain threshold; the ratio between $I(cb)$ and $I_{MS}(cb)$ should be greater than 1; and the ratio between $I(cr)$ and $I_{MS}(cr)$ should be less than 1 for an eye candidate. For the skin color, the previous two ratios should all be very close to 1. The eye candidates in an image can therefore be determined as follows:

**Fig. 6** The training set for upper face.

**Fig. 7** The training set for lower face.

$$P_y(y) = \begin{cases} 1 & I(y) < t_y \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

$$P_{cb}(cb) = \begin{cases} 1 & \dfrac{I(cb)}{I_{MS}(cb)} > t_{cb} \\ 0 & \text{otherwise} \end{cases}, \quad \text{and}$$

$$P_{cr}(cr) = \begin{cases} 1 & \dfrac{I(cr)}{I_{MS}(cr)} < t_{cr} \\ 0 & \text{otherwise} \end{cases},$$

where $P_y(y)$, $P_{cb}(cb)$, and $P_{cr}(cr)$ represent a possible eye candidate when the Y, Cb, and Cr components are used, respectively, and the thresholds $t_y$, $t_{cb}$, and $t_{cr}$ are the corresponding thresholds for the Y, Cb, and Cr components. Therefore, possible eye candidates eye($x$) can be determined as follows:

$$\text{eye}(x = \{y, cb, cr\}) = P_y(y) \cap P_{cb}(cb) \cap P_{cr}(cr) \cap P_{\text{face}}(x). \tag{5}$$

In this equation, we confine the detection within the segmented skin-color regions $P_{\text{face}}(x)$ from Eq. (3). The possible eye candidates eye($x$) are illustrated in Fig. 4(c).

From Fig. 4(c), it can be observed that the possible eye candidates are located around the face contour and hair, and some in the face region. Further elimination of these eye candidates is required, so that the computational complexity of searching the possible face candidates can be reduced. We observe that the eye has a strong horizontal edge, while the face contour has a strong vertical edge when the face orientation is between −45 and +45 deg. The edge map of a face image is generated using the Sobel edge detector, but only with its luminance component. In our approach, whenever a possible eye candidate has a strong horizontal edge intensity and a weak vertical edge intensity, it will not be removed. In addition, the skin color is yellowish, and the eye is white and dark. Therefore, the difference in the red component should be large. At each possible eye candidate location, the difference in the red component is measured in their $3 \times 3$ neighborhood. Without removing the local information, another $3 \times 3$ search window is located at each possible eye candidate to group the surrounding candidates into one. These properties are used to reduce the number of possible eye candidates and results are shown in Fig. 5.

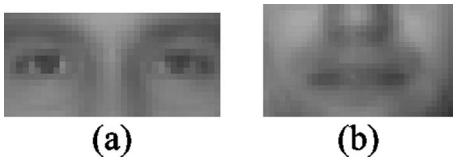

**Fig. 8** Face templates: (a) upper face template and (b) lower face template.

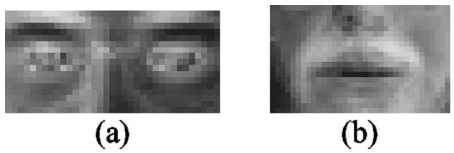

**Fig. 9** The eigenmasks: (a) eigenmask for the upper face and (b) eigenmask for the lower face.

Since the size of a human face is proportional to the distance between its two eyes, a possible face region containing the eyebrows, eyes, nose, and mouth can be formed based on this relationship. In our approach, a square block is used to represent possible face candidates. The size of the square block is determined by means of the head model.[10] Based on the head model and the locations of the possible eye pairs, a population of possible face regions with different locations, sizes, and orientations are generated by pairing the possible eye candidates.

Once a face candidate is selected, its orientation and size are normalized. Then, histogram normalization[18] is applied to the Y component of the selected face candidate to compensate for nonuniform lighting; this can help improve detection reliability and accuracy. In this normalization process, the histogram of a possible face region is transformed into the histogram of a reference face image for approximating a similar illumination condition. The method is effective, since all faces have the same structure, similar shape, and illumination properties. Finally, the image is passed to the final stage for further verification.

### 2.3 *Two-Step Face Verification Using Eigenmask*

To determine whether the normalized face candidate is a face or not, the similarity between the face candidate and a face template is measured. In our approach, a two-step face verification procedure is performed. Instead of using a single face template, a face region is separated into two parts: the upper part contains the eyes, while the lower part contains the nose and mouth. This can make the similarity measure more localized. A true face will be declared only if a face region has both its upper and lower parts similar to the corresponding two face templates. In our algorithm, the

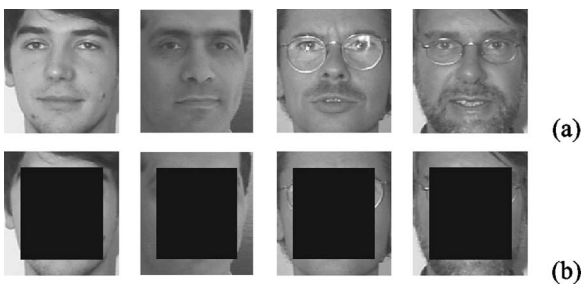

**Fig. 10** Examples of face boundary regions: (a) original face and (b) face boundary regions extracted from the original faces.

**Table 1** Detection performance based on the HHI MPEG-7 face database.

| Lighting condition | Overhead lights | Dark or side lights | Strong overhead lights | Strong side lights | Total |
|---|---|---|---|---|---|
| Number of images | 54 | 49 | 22 | 26 | 151 |
| Detection method | Pixel-by-pixel skin-color segmentation and no face boundary verification | | | | |
| Number of correctly detected faces | 53 | 43 | 16 | 17 | 129 |
| Number of missed face | 1 | 5 | 3 | 7 | 16 |
| Number of false alarms | 0 | 1 | 3 | 2 | 6 |
| Detection rate (%) | 98.1% | 87.7% | 72.7% | 65.4% | 84.8% |
| Detection method | Region-based skin-color segmentation and no face boundary verification | | | | |
| Number of correctly detected faces | 53 | 46 | 18 | 21 | 138 |
| Number of missed face | 1 | 2 | 2 | 3 | 8 |
| Number of false alarms | 0 | 1 | 2 | 2 | 5 |
| Detection rate (%) | 98.1% | 93.5% | 81.8% | 80.8% | 91.3% |
| Detection method | Region-based skin-color segmentation with face boundary verification | | | | |
| Number of correctly detected faces | 54 | 46 | 19 | 21 | 140 |
| Number of missed face | 0 | 2 | 2 | 3 | 7 |
| Number of false alarms | 0 | 1 | 1 | 2 | 4 |
| Detection rate (%) | 100% | 93.5% | 86.4% | 80.8% | 92.7% |

nose and mouth form the lower part of our face template, while the eyes form the upper part. All the training images used to construct the face templates are normalized to a specific size.

Both the upper and lower templates are grayscale images, and are obtained by calculating the average of a set of preprocessed training face images. The training set contains face images of 14 males and 6 females, different races, ages, with and without glasses, and a mustache. The training images and the corresponding face template are shown in Figs. 6 and 8(a) for the upper faces, and Figs. 7 and 8(b) for the lower faces.

The distance between a possible face region and the corresponding face template can be measured by means of the Euclidean distance with a certain weighting function based on the importance of the human facial features. The weighting function can be obtained from the eigenmask,[12] denoted as $E_{\text{mask}}$. The eigenmask is generated from the first eigenface[19] of the training face images. The first eigenface represents the principal components of the training face images, which have larger magnitudes at important locations such as the eyes, nose, and mouth. Figure 9 shows the eigenmasks generated whose values are normalized between 0 and 255, and which are used as a weighting function in the distance measurement. As shown in Fig. 9, the eye, nose, and mouth regions have higher magnitudes than other facial regions and those with glasses.

The use of the eigenmasks can reduce false alarms and increase the distance when comparing a nonface region to

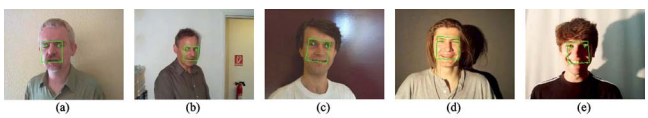


**Fig. 11** Face detection results with the HHI MPEG-7 face database: (a) frontal view faces under overhead lights, (b) near frontal view faces under overhead lights, (c) faces under dark or side lights, (d) faces under strong overhead lights, and (e) faces under strong overhead side lights.


**Fig. 12** Face detection results with the AR face database: (a) frontal view male faces with different facial expressions under overhead lights, (b) frontal view male faces under side lights, (c) frontal view female faces with different expressions under overhead lights, and (d) frontal view female faces under side lights.

**Table 2** Detection performance based on the AR face database.

| Lighting condition<br>Gender | Overhead lights<br>Male | Strong side lights<br>Male | Overhead lights<br>Female | Strong side lights<br>Female | Total |
|---|---|---|---|---|---|
| Number of images | 298 | 223 | 231 | 168 | 920 |
| Detection method | Region-based skin-color segmentation and no face boundary verification | | | | |
| Number of correctly detected faces | 297 | 205 | 231 | 150 | 883 |
| Number of missed face | 1 | 18 | 0 | 18 | 37 |
| Number of false alarms | 0 | 0 | 0 | 0 | 0 |
| Detection rate (%) | 99.7% | 91.9% | 100% | 89.0% | 96.0% |
| Detection method | Region-based skin-color segmentation with face boundary verification | | | | |
| Number of correctly detected faces | 297 | 211 | 231 | 159 | 898 |
| Number of missed face | 1 | 12 | 0 | 9 | 22 |
| Number of false alarms | 0 | 0 | 0 | 0 | 0 |
| Detection rate (%) | 99.7% | 94.6% | 100% | 94.6% | 97.6% |

the upper and lower face templates. The distance measure between an input candidate and the template is given as follows:

$$\varepsilon = \frac{1}{MN}\sum_{y=0}^{M}\sum_{x=0}^{N} E_{\text{mask}}(x,y)\cdot[I(x,y)-F_{\text{avg}}(x,y)]^2, \tag{6}$$

where $I$ is the luminance component of the possible face candidate, $F_{\text{avg}}$ is the average of the training set, and $MN$ is the size of $I$. The face candidate will be declared a true face if the values $\varepsilon_{\text{upper}}$ and $\varepsilon_{\text{lower}}$ for the upper and lower face parts are both smaller than the thresholds $T_{\text{upper}}$ and $T_{\text{lower}}$, respectively. Therefore, a number of faces can be detected in an image at this stage. In our algorithm, as the upper region is more important, we compute the upper part first. The calculation of the lower face will be performed if the upper face has been verified. Therefore, the computation required can be reduced. For overlapping regions, the one with the lowest value of

$$\varepsilon_{\text{total}} = \frac{\varepsilon_{\text{upper}} + \varepsilon_{\text{lower}}}{2}, \tag{7}$$

will be chosen as the true face region.

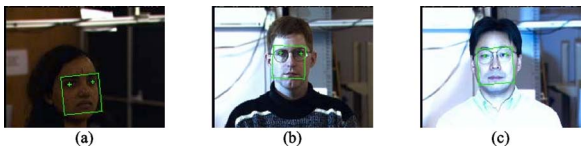


**Fig. 13** Face detection results for frontal and near-frontal view with the CMU PIE database: (a) dark lights, (b) side lights, and (c) strong overhead lights.

### 2.4 *Face Boundary Verification*

Once the face has been verified and selected in the previous verification process, the selected faces are further verified based on the appearance of their face boundaries to reduce the false alarm. Some face boundary regions, as shown in Fig. 10, are first extracted, and feature vectors $x$ are formed. The black rectangles in Fig. 10 represent face regions and are not included in the feature vector.

Since the number of training face boundaries is much smaller than the dimensionality of the corresponding feature vectors, the covariance matrix is singular in practice. Thus, the face boundary vector $x$ is first projected onto a lower-dimensional subspace by means of principal component analysis.[2] The low-dimensional feature vector $y = \mathbf{E}_M^T \tilde{x}$, where $\tilde{x} = x - \bar{x}$ and $\mathbf{E}_M$ is a matrix containing $M$ column vectors, which are the eigenvectors of the corresponding covariance matrix with the largest eigenvalues. The face boundary region can be modeled by a Gaussian density function:

$$P(y) = \frac{\exp\left[-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)\right]}{(2\pi)^{M/2}|\Sigma|^{1/2}}, \tag{8}$$

where $P(y)$ is an $M$-dimensional Gaussian density function with mean vector $\mu$ and covariance $\Sigma$ of the low-

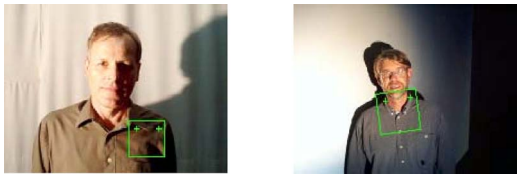

**Fig. 14** Examples of false alarms.

**Table 3** Detection performance based on the CMU PIE database.

| Lighting conditions | Dark | Strong side lights | Strong overhead lights | Total |
|---|---|---|---|---|
| Number of images | 408 | 1360 | 1496 | 3264 |
| Detection method | Pixel-by-pixel skin-color segmentation and no face boundary verification | | | |
| Number of correctly detected faces | 353 | 965 | 1163 | 2481 |
| Number of missed face | 41 | 385 | 316 | 742 |
| Number of false alarms | 14 | 10 | 17 | 41 |
| Detection rate (%) | 86.5% | 70.9% | 77.7% | 76.0% |
| Detection method | Region-based skin-color segmentation and no face boundary verification | | | |
| Number of correctly detected faces | 362 | 1099 | 1287 | 2748 |
| Number of missed face | 35 | 255 | 196 | 486 |
| Number of false alarms | 11 | 6 | 13 | 30 |
| Detection rate (%) | 88.7% | 80.8% | 86.0% | 84.2% |
| Detection method | Region-based skin-color segmentation with face boundary verification | | | |
| Number of correctly detected faces | 371 | 1169 | 1318 | 2858 |
| Number of missed face | 30 | 187 | 167 | 384 |
| Number of false alarms | 7 | 4 | 11 | 22 |
| Detection rate (%) | 90.9% | 86.0% | 88.1% | 87.6% |

dimensional training samples. In our algorithm, the width of the face boundary is five pixels at each of the four borders. As the boundary regions mainly contain the face contours only, a Gaussian density function is sufficient to represent the appearance of the face boundary, rather than using a mixture-of-Gaussians model. With a face candidate verified by the two eigenmasks, its boundary region is projected onto the eigenspace to form a low-dimensional vector. Then the likelihood $P(y)$ of the region being a face boundary is measured by using Eq. (8). If $P(y)$ is larger than a certain threshold, the face candidate will be considered a true face.

## 3 Experimental Results

Experiments were carried out to evaluate the detection performance of our algorithm using the HHI MPEG-7 face database,[20] the AR face database,[21] and the CMU pose, illumination, and expression (PIE) face database.[22] The aim of our proposed method is to detect the frontal or near-frontal view of faces under varying lighting conditions and facial expressions. In our experiments, a face is considered to be correctly detected if the detected position of the two eyes is exactly matched. A face is said to be missed if the face region can be located but the two eye positions are mismatched. A detection is said to be a false alarm if the true face region is not covered.

The HHI MPEG-7 face database contains 206 images, each $640 \times 480$ pixels in size. The database contains people of different races, under varying lighting conditions, and from frontal view to profile file. Thus, 151 images with varying lighting conditions were selected, and those with profile views were ignored in the experiment. The detection performance is tabulated in Table 1 and results are shown in Fig. 11. Using the traditional pixel-by-pixel skin-color segmentation, the detection rate is 85.4% without face boundary verification. By using our proposed region-based skin-color segmentation, the overall detection rates are 92.7 and 91.3% with and without the use of face contour verification, respectively. To investigate the performance of our algorithm under different lighting conditions, we have classified the face images according to the lighting conditions;

**Fig. 15** Examples of missed faces.

**Fig. 16** Examples of multiple-face detection.

these include overhead lights, dark or side lights, strong overhead lights, and strong side lights. The corresponding detection rates are 100, 93.5, 86.4, and 80.8%, respectively.

The AR face database contains 126 faces, with 70 men and 56 women, each $768 \times 576$ pixels in size. The face images in this database are frontal views, but have different facial expressions, illumination conditions, and occlusions. We selected 920 images with varying illumination and facial expressions from this database. The images with occlusion are ignored. The detection performance is tabulated in Table 2 and results are shown in Fig. 12. Since the images in the AR face database do not have background, only the region-based skin-color segmentation is employed. The overall detection rates are 97.6 and 96.0% with and without using face contour verification, respectively. We have classified face images according to gender and lighting conditions (overhead lights and strong side lights). The corresponding detection rates for male face images are 99.7 and 94.6%, and 100 and 94.6% for females, respectively.

The CMU pose, illumination, and expression (PIE) database contains 41,368 images of 68 people, and each image is $640 \times 486$ pixels in size. There are 13 different poses, 43 different illumination conditions, and four different expressions for each person. We selected 3264 images that contain frontal and near frontal faces with different genders, different races, and under varying lighting conditions. With the traditional pixel-by-pixel skin-color segmentation, the detection rate is 76.0% without using face boundary verification. Using our proposed region-based skin-color segmentation, the overall detection rates are 87.6 and 84.2% with and without the use of face boundary verification, respectively. The detection performance is tabulated in Table 3 and results are shown in Fig. 13. We have classified the face images according to different lighting conditions, including dark lights, strong side lights, and strong overhead lights. The corresponding detection rates are 90.9, 86.0, and 88.1%, respectively.

The experiments were conducted on a Pentium IV 1.7-GHz computer. The average processing time for locating faces in a picture ranges from 0.3 to 1.5 s. The experiments show that our method can achieve a high detection rate irrespective of lighting conditions and facial expressions. When the face boundary is also considered, the detection rate is improved and false alarms are reduced. For the HHI database, our algorithm can achieve a detection rate of 92.7% compared to 88.89% in Hsu, Abdel-Mottaleb, and Jain[8] and 90.07% for frontal and near-frontal faces in Wong, Lam, and Siu.[9] Missed faces were mainly due to severely poor lighting conditions at the eye regions, or strong yellow lighting projected on the clothes and hair, which then assume a skin-like color. Some false alarms and cases of missing faces are shown in Figs. 14 and 15. As our algorithm can also detect multiple faces in an image, some examples of the detection are shown in Fig. 16. In conclusion, our method can achieve a fast and high face-detection rate under varying lighting conditions and facial expressions.

## 4 Conclusion

We propose a reliable face detection approach under varying lighting conditions. In our algorithm, a color compensation scheme is adopted to alleviate the effect of strong lighting conditions on skin color. Then we consider the distributions of the skin color of segmented regions under different lighting. Based on the color information, possible eye candidates are detected within the face-like regions, and possible face candidates are then formed by pairing two possible eye candidates in a face-like region. A two-step eigenmask verification process is proposed, with a weighting function used to measure the distance between a face candidate and the face template.

Finally, boundary regions of the face candidates are verified with a Gaussian density function to reduce false alarms. Experimental results show that our algorithm can achieve a higher detection rate and reduce the number of false alarms. Furthermore, our method can detect faces of different sizes and orientations under varying lighting conditions and different facial expressions.

### Acknowledgment

The work described was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (project number PolyU5123/01E) and by a research grant from The Hong Kong Polytechnic University, Hong Kong, China.

### References

1. M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detection faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(1), 34–58 (2002).
2. B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7), 696–710 (1997).
3. C. Liu, "A Bayesian discriminating features method for face detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(6), 725–740 (2003).
4. C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.* **38**(1), 15–33 (2000).
5. A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(4), 349–361 (2001).
6. H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(1), 22–38 (1998).
7. D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone application," *IEEE Trans. Circuits Syst. Video Technol.* **9**(4), 551–564 (1999).
8. R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 696–706 (2002).
9. K. W. Wong, K. M. Lam, and W. C. Siu, "A robust scheme for live detection of human faces in color images," *Signal Process. Image Commun.* **18**, 103–114 (2003).
10. K. W. Wong, K. M. Lam, and W. C. Siu, "An efficient algorithm for human face detection and facial feature extraction under different conditions," *Pattern Recogn.* **34**, 1993–2004 (2001).
11. H. Greenspan, J. Goldberger, and I. Eshet, "Mixture model for face-color modeling and segmentation," *Pattern Recogn. Lett.* **22**, 1525–1536 (2001).
12. K. W. Wong, K. M. Lam, and W. C. Siu, "An efficient color compensation scheme for skin color segmentation," *IEEE Int. Symp. Circuits Syst. (ISCAS2003)* **2**, 480–483 (2003).
13. D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 1–18 (2002).
14. Y. Wang and B. Yuan, "A novel approach for human face detection from color images under complex background," *Pattern Recogn.* **34**, 1983–1992 (2001).
15. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 564–575 (2003).
16. T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.* **13**, 47–60 (1996).
17. K. M. Lam, "A fast approach for detecting human faces in a complex background," *Proc. IEEE Intl. Symp. Circuits Syst.* 4, 85–88 (1998).
18. P. J. Phillips and Y. Vardi, "Efficient illumination normalization of facial images," *Pattern Recogn. Lett.* **17**, 921–927 (1996).
19. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn Neurosci.* **3**(1), 71–86 (1991).

20. "MPEG7 content set from Heinrich Hertz Institute," see http://www.darmstadt.gmd.de/mobile/hm/projects/MPEG7/Documents/N2466.hmtl (1998).
21. A. M. Martinex and R. Benavente, "The AR face database," *CVC Technical Report 24* (1998).
22. T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(12), 1615–1618 (2003).

**Tze-Yin Chow** received his BEng and MPhil degrees in electronic and information engineering from The Hong Kong Polytechnic University, Hong Kong, in 2002 and 2005, respectively. He is currently working at SiS International Limited as a product specialist. His research interests include motion tracking, human face detection, and human face coding.

**Kin-Man Lam** received his Associateship in electronic engineering with distinction from The Hong Kong Polytechnic University, formerly called Hong Kong Polytechnic, in 1986. He won the S.L. Poa Scholarship for overseas studies and was awarded a MSc degree in communication engineering from the Department of Electrical Engineering, Imperial College of Science, Technology and Medicine, England, in 1987. In August 1993, he undertook a PhD degree program in the Department of Electrical Engineering at the University of Sydney, Australia, and won an Australia Postgraduate Award for his studies. He completed his PhD studies in August 1996, and was awarded the IBM Australia Research Student Project Prize. He joined the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University as an assistant professor in 1996, and has been an associate professor since 1999. Dr Lam has also been a member of the organizing committee and program committee of many international conferences. His current research interests include human face recognition, image and video processing, and computer vision.

**Kwok-Wai Wong** received his BEng degree and MPhil degree in electronic and information engineering from The Hong Kong Polytechnic University, Hong Kong, in 1998 and 2001, respectively. He is currently a research associate in the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His research interests include human face detection, image and video processing, pattern recognition with applications in computer vision, and inspection systems.