MEAN-SHIFT BASED MIXTURE MODEL FOR FACE DETECTION IN COLOR IMAGE

Tze-Yin Chow and Kin-Man Lam

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering,
The Hong Kong Polytechnic University, Hong Kong.

ABSTRACT

Human face detection is a challenging task under different lighting conditions. In this paper, we propose an efficient and reliable algorithm to detect human faces in an image. In our algorithm, skin-colored pixels under various lighting conditions are identified by using a region-based approach. Within the detected skin-color regions, a ratio method is proposed to determine possible eye candidates. Two eye candidates form a possible face region, which is then verified by means of a two-stage procedure with an eigenmask. Experimental results based on the HHI MPEG-7 face database show that this face detection algorithm is efficient and reliable under different lighting conditions.

1. INTRODUCTION

Face detection is the first step in any face processing system. This is a challenging task because the human face is highly variable. The detection performance may be affected by the presence of eyeglasses, different races, genders, facial hair, facial expressions, lighting conditions, etc. Numerous approaches [5-7] have been proposed for the detection of human faces in gray-level images. All these face searching techniques also slide an $m \times n$ observation window with multi-scales throughout the image to perform matching. Another method in [4] uses eyes and mouth color properties and then a triangular relationship is formed for face detection. The computation complexities of these methods are usually too high for real-time applications.

In this paper, we propose a new method for detecting and locating human faces in color images under various lighting conditions. The detection process consists of three steps. The first step is to segment the color image by using the mean-shift algorithm [2]. The segmented regions are then processed by a color compensation scheme [10], and skin-color is identified by means of the maximum-likelihood method [3]. In the second step, possible eye candidates are located 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 is adopted to verify the possible face candidates. Experiments were carried out with the HHI MPEG-7 face database.

2. FACE COLOR SEGMENTATION

In order to segment human faces in a complex background, skin color information is a commonly used technique. In [1], a high detection rate can be achieved if the face images are captured under good lighting conditions. However, some skin pixels cannot be located properly under poor lighting conditions. We therefore propose a robust color compensation [10] method with the use of a mixture-of-Gaussian model [3] to represent skin color under various illuminations to solve this problem.

2.1 Color Image Segmentation with Mean Shift

Mean-Shift Algorithm [2] is a kernel-based density estimation technique which has been used in many applications including data clustering, image segmentation, object tracking, etc. In particular, it is a nonparametric and robust technique to analyze feature spaces.

Given n training color vectors x_i, $i=1, \ldots, n$, in the d-dimensional space R^d. The feature space can be modeled by an unknown kernel density function K

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

where x^s and x^r are the spatial part and the color part, respectively, 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:

$$\mathrm{m}(x) = \frac{\sum_{i=1}^{n} x_i K(x - x_i)}{\sum_{i=1}^{n} K(x - x_i)}. \qquad (2)$$

The vector, x, is updated in the form of iteration such that $x \leftarrow \mathrm{m}(x)$ with $\mathrm{m}(x)=\{\mathrm{m}(x);\ x \in R^d\}$. The difference $\mathrm{m}(x)-x$ is called the mean shift, and the process iterates until converged with zero gradient, i.e. $\mathrm{m}(x)-x=0$. 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.

The kernel used in our algorithm is the Epanechnikov kernel [2], which provides a similar performance and simpler structure to the Gaussian kernel. The Epanechnikov kernel profile is defined as follows:

$$k_E(x) = \begin{cases} 1 - \|x\|^2 & 0 \le x \le 1, \\ 0 & x > 1. \end{cases} \qquad (3)$$

The color image segmentation algorithm consists of two steps.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

1. Mean shift filtering which smoothes the input image by running the mean shift algorithm.
2. Mean shift segmentation which delineates the smoothed image and purges the small regions.

2.2 Face Color Modeling

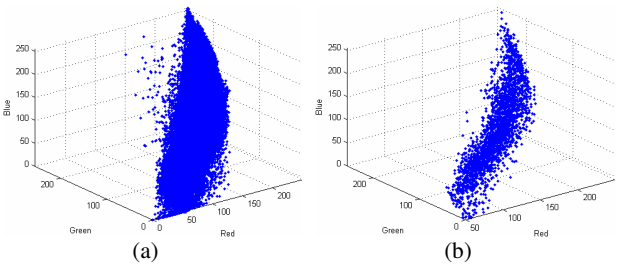
A skin-color model can be trained by learning from a large set of sample skin-color pixels. Under varying illumination, the skin color distribution is no longer unimodal. To tackle skin pixels under various lighting conditions, the Gaussian mixture model is used.

In our approach, the training skin-color samples are obtained from segmented face regions. The face regions are extracted from the HHI MPEG-7 face database. This can ensure that there is no outlier skin color in the training set, and each face image is segmented with the mean-shift algorithm. After segmentation, a face region is partitioned into a number of regions, and the color used to represent a region is the mode of its color pixels. Some face images and their corresponding segmented results under various lighting condition are shown in Fig. 1.

(a) (b) (c) (d) (e) (f)

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

A total of 366,431 skin-color pixels were extracted from 206 faces. After color image segmentation using the mean-shift algorithm, 2,425 skin-color regions were produced. The average number of regions in each segmented face is 11.7. The modes of the skin-color regions are then used to represent skin color. 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 the colors other than the skin color, such as eye ball, eyebrow, mustache, lip, etc. In Fig. 2(b), there are only a few pixels apart from the skin color distribution. After segmentation, the outlier colors are fused into the skin color domain, as shown in Fig. 1. All of the eyes, mouth and eyebrows are removed and segmented to become skin-like colors.



(a) (b)

Fig. 2. RGB distribution of the face color. (a) RGB distribution of original faces. (b) RGB distribution of segmented faces.

2.3 Face Color Modeling via Gaussian Mixture Model

Before applying the Gaussian mixture model to describe the distributions of the segmented face colors in Fig. 2(b), the *K*-Means algorithm is applied to cluster the face colors to obtain the initial parameters and the number of clusters for the Gaussian mixture model. Each cluster represents the skin color under different lighting conditions. Since the RGB color space does not contain any lighting information, the face color is first translated to the compensated YCbCr color space [10]. The face color is then divided into k+1 decision regions; k face-color regions and the complementary non-face region. In our approach, we set k = 5, which can empirically segment the skin and non-skin color well throughout our experiments.

With the results from the *K*-Means algorithm, the skin-color distribution in the compensated YCbCr color space can be modeled using the Gaussian mixture model [3]. In order to optimize the mixture parameters, maximum-likelihood can be used to seek the best parameters based upon the results from the *K*-Means algorithm. Then, the Expectation-Maximization (EM) algorithm [3] is used to estimate the many-to-one mapping from the Gaussian mixture model. After the optimal parameters of the mixture model are computed, a mapping decision in [3] is used to distinguish the skin and non-skin colors.

2.4 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 at the eye, nose and mouth regions. These holes can be removed by using the morphological open and close operations. However, some big holes in the background may not be removed, as shown in Fig. 3(b). The result will become worse when the image concerned has uneven lighting condition, or is under a strong illumination effect.

Because of the drawback in the pixel-by-pixel approach, we propose performing skin-color segmentation by using a region approach. In our algorithm, a color image is first segmented by the mean-shift algorithm, and a result is shown in Fig. 3(c). As a result, the image is segmented into many regions, and each region is represented by the mode of the color pixels. The mapping decision in [3] is then applied to determine whether the region is of skin color or not. If a mode is classified as a skin color, the whole region will be declared a skin-color region. The region-based approach can achieve a better performance level as compared to the pixel-by-pixel method under uneven lighting conditions, as shown in Fig. 3(d).

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

Fig. 3. Face color segmentation under uneven light conditions: (a) original image; (b) face color segmentation with pixel-by-pixel approach; (c) color image segmentation with mean-shift algorithm; and (d) face color segmentation with region-based approach.

3. POSSIBLE EYE CANDIDATE DETECTION

An efficient way to detect an eye region is by means of valley detection [9]. However, under poor lighting conditions, valley detection may fail, especially when the eye regions are

shadowed. In this section, an improved method is presented to detect eye regions more reliably under various lighting conditions.

Since the eyes are surrounded by skin, there is a significant color distance 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 properties will be used to determine possible eye candidates in a skin-color region.

In our approach, an image is first segmented with the mean-shift algorithm, denoted as $I_{MS}(c)$, where c=(y, cb, cr). In the segmented image, as shown in Fig. 4(b), the details of the eye are removed and replaced by skin-like color.

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 as $I_{MS}(y)$, $I_{MS}(cb)$, and $I_{MS}(cr)$ for 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 above ratios should all be very close to 1. The eye candidates in an image can therefore be determined as follows:

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

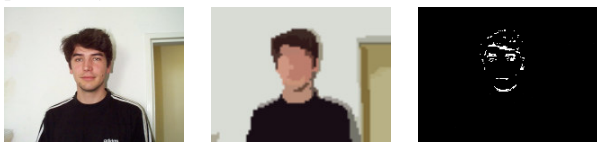
$$P_{cb}(cb)=\begin{cases}1 & \dfrac{I(cb)}{I_{MS}(cb)}>t_{cb}\\ 0 & \text{otherwise}\end{cases},$$

$$P_{cr}(cr)=\begin{cases}1 & \dfrac{I(cr)}{I_{MS}(cr)}>t_{cr}\\ 0 & \text{otherwise}\end{cases}, \qquad (4)$$

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 respectively. Therefore, possible eye candidates, $Eye(x)$, can be determined as follows:

$$Eye(x=\{y,cb,cr\})= P_y(y)\cap P_{cb}(cb)\cap P_{cr}(cr)\cap P_{face}(x) \qquad (5)$$

In the above equation, we confine the detection within the segmented skin-color regions, $P_{face}(x)$. Fig. 4(c) illustrates the possible eye candidates, $Eye(x)$.



(a) (b) (c)

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

From Figure 4(c), some possible eye candidates are located around the face contour and hair, and some in the face region. Further reduction is required to reduce the computational complexity in the face verification process. We observe that the eye has a strong horizontal edge, while the face contour has a strong vertical edge. In addition, the skin color is yellowish, and the eye is white and dark. Therefore, the difference in red component should be large. These properties are used to reduce the number of possible eye candidates.

The edge map of the luminance component of a face image is generated using the Sobel edge detector. The horizontal edge and vertical edge are denoted as $S_H(y)$ and $S_V(y)$, respectively, where y is the luminance component. The set of selected possible eye candidates, $Eye'(x)$, is formed as follows:

$$Eye'(x)=(S_H(y)<T_{SH})\cap(\overline{S_V(y)<T_{SV}})\cap Eye(x), \qquad (6)$$

where T_{SH} and T_{SV} are the thresholds for the horizontal and vertical edge intensities. In (6), whenever a possible eye candidate has a strong horizontal edge intensity and a weak vertical edge intensity, it will not be removed. A 3×3 window is located at the possible eye candidates to measure the difference in the red component in their neighborhood. A possible eye candidate will be removed if

$$\max(R)-\min(R)<T_{red}, \qquad (7)$$

where R is the neighborhood red component and T_{red} is a threshold. The possible eye candidates are removed by using (6) and (7), and the results are shown in Fig. 5(b). In Fig. 5(b), the remaining possible eye candidates are connected to each other. Further elimination can be done without removing the local information. A 3×3 searching window is located at each possible eye candidate to group the surrounding candidates into one. The result is shown in Fig. 5(c).

(a) (b) (c)

Fig. 5. Possible Eye Candidate Reduction: (a) possible eye candidates; (b) reduction of possible eye candidates by Sobel and red color distance measurement; and (c) possible eye candidates in (b) grouped by 3×3 window.

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 possible face candidate is represented by a square block according to the head model in [9]. A population of possible face regions with different locations, sizes, and orientations are generated by pairing the possible eye candidates, as shown in Fig. 6. Then, histogram normalization [8] is applied to the selected face candidates to compensate for non-uniform lighting; this can help improve detection reliability and accuracy. Finally, the image will be passed to the final stage for further verification.

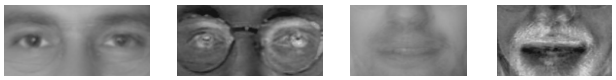
Fig. 6. Selection of possible face regions.

4. A TWO-STEP FACE VERIFICATION USING EIGENMASK

In order 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. Facial features are localized in the similarity measurement. 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 nose and mouth form the lower part of our face template, while the eyes form the upper part. All the training images, which are selected from the HHI MPEG-7 face database, used to construct the face templates are normalized to a specific size.

Both the upper and lower templates are gray-scale images, and the face templates are obtained by calculating the average of a set of pre-processed training face images, as shown in Figures 7(a) and 7(c). The training set contains face images of different races, ages, with and without eyeglasses and a moustache.



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

Fig. 7. (a) Upper face template. (b) Upper face eigenmask. (c) Lower face template. (d) Lower face eigenmask.

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 in [11]. The eigenmask is generated from the first eigenface of the training face images, as shown in Figures 7(b) and 7(d). The use of the eigenmask can reduce false alarms and increase the distance when comparing a non-face region to the upper and lower face templates. 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 smallest distance will be chosen as the true face region.

5. EXPERIMENTAL RESULTS

The detection performance was evaluated based on using the HHI MPEG-7 face database. The aim of our proposed method is to detect the frontal and near-frontal view of faces under varying lighting conditions and facial expressions. A face is said to be detected if the two eyes are exactly matched.

The HHI MPEG-7 face database contains 206 face subjects of different races, and under varying lighting conditions ranging from dark and shadow to strong overhead-projected, 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 overall detection performance of our approach is 90.1%. We have also classified the faces according to the lighting conditions; these include overhead lights, dark or side lights, strong overhead lights, and strong side lights. The corresponding detection rates are 98.1%, 91.8%, 78.3%, and 80.8%; and the false alarm rates are 0.0%, 2.0%, 9.1% and 7.7%, respectively.

The experiments were conducted on a Pentium IV 1.7GHz computer. The average processing time for locating faces in a picture ranges from 0.5s to 2.8s. The experiments show that our method can achieve a high detection rate irrespective of lighting conditions as compared to 88.9% in [11] and 80.58% in [4].

6. CONCLUSION

In this paper, we have proposed a more reliable face detection approach under varying lighting conditions. In our algorithm, the distributions of skin color under different lighting are modeled. Within the face-like regions, possible eye candidates are detected. Then, possible face candidates are formed by pairing two possible eye candidates. Finally, a two-step eigenmask verification process is proposed to measure the distance between a face candidate and the face template. Experimental results show that our algorithm can achieve high detection rate and can reduce the number of false alarms. Furthermore, our method can detect faces of different sizes under varying lighting conditions.

7. REFERENCE

[1] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone application," *IEEE Trans. on Circuits and System for Video Technology*, vol. 9, pp. 551-564, 1999.

[2] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1-18, 2002.

[3] H. Greenspan, J. Goldberger, and I. Eshet, "Mixture model for face-color modeling and segmentation," *Pattern Recognition Letters*, vol. 22, pp. 1525-1536, 2001.

[4] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 696-706, 2002.

[5] C. Liu, "A bayesian discriminating features method for face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 725-740, 2003.

[6] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 696-710, 1997.

[7] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 349-361, 2001.

[8] P. J. Phillips and Y. Vardi, "Efficient illumination normalization of facial images," *Pattern Recognition Letters*, vol. 17, pp. 921-927, 1996.

[9] 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 Recognition*, vol. 34, pp. 1993-2004, 2001.

[10] K.-W. Wong, K.-M. Lam, and W.-C. Siu, "An efficient color compensation scheme for skin color segmentation," presented at IEEE International Symposium on Circuits and systems (ISCAS2003), Bangkok, Thailand, 2003.

[11] K.-W. Wong, K.-M. Lam, and W.-C. Siu, "A robust scheme for live detection of human faces in color images," *Signal Processing: Image Communication*, vol. 18, pp. 103-114, 2003.