Pattern Recognition 43 (2010) 1413–1420

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.de/locate/pr

Skin detection for single images using dynamic skin color modeling

Hung-Ming Sun *

Department of Computer Science and Information Engineering, Kainan University, No. 1 Kainan Road, Luchu, Taoyuan County 33857, Taiwan, ROC

ARTICLE INFO

Article history:
Received 22 August 2008
Received in revised form
17 August 2009
Accepted 21 September 2009

Keywords:
Skin detection
Adaptive skin color model
Clustering algorithm
Bayes classifier

ABSTRACT

Up-to-date skin detection techniques use adaptive skin color modeling to overcome the varying skin color problem. Most methods for tracking skin regions in videos utilize the correlation between contiguous frames. This paper proposes a new approach for detecting skin in a single image. This approach uses a local skin model to shift a globally trained skin model to adapt the final skin model to the current image. Experimental results show that the proposed method can achieve better accuracy. Two improvements for speeding up the processing are also discussed.

1. Introduction

Skin detection is a common preprocessing step for analyzing images or videos of human beings. It aims to track the human body for further recognition. Its application includes face detection and recognition, gesture analysis, image content classification, etc. Earlier studies [1–3] have shown that human skin color generally clusters within a certain range in the color space and can be used to identify skin pixels in an image. Skin detection methods reported in the literature include explicit skin-color thresholding, Bayes classifiers, neural network classifiers, maximum entropy classifiers, and so on [4]. The explicit skin-color thresholding technique [5–7] is very fast and requires little storage space because it only needs a few decision rules for classification. However, the system lacks flexibility as the decision rules are stiffly fixed. Neural network methods can acquire decision boundaries from the training data [8–10] but the training stage may take a long time if the number of training patterns is very large (for example, there are nearly two billion pixels in our skin/non-skin image database). Zheng et al. [11] employed a maximum entropy model to detect skin in images, which impose constraints on the color gradients of neighboring pixels and use a Bethe tree approximation [12] and a belief propagation algorithm to calculate the skin probability at each pixel. Their method has to deal with six color histograms owing to the involvement of the neighboring pixels in the process.

Compared with the above methods, the Bayes classifier provides another solution especially suitable for extremely large amounts of training data. The decision strategy of a Bayes classifier is based on calculating the likelihood functions of two classes. Because the likelihood functions of the skin class and the non-skin class correspond to the color histograms of the two classes and can be calculated in a single pass, its training is very fast. Jones et al. [1] collected 18 696 photographs from the Internet, containing nearly two billion pixels. By manually separating skin and non-skin images and labeling the skin pixels, they derive a skin and a non-skin color histogram from these images in the red green blue (RGB) color space. Then, the ratio of skin likelihood to non-skin likelihood for a given color can be calculated by examining the skin and non-skin color histograms and the result is compared with a threshold to determine whether the color is skin or not. The execution speed of this method is very fast because it needs only a little calculation. Some other skin detection systems also adopt similar approaches [13–15]. However, such methods (referred to as nonparametric modeling) require some memory space for storing the two histograms. To reduce the memory requirement and also obtain well-generalized distribution models from less training data, the Gaussian function can be used to represent the skin and non-skin color models (referred to as parametric modeling) [16–19]. Regardless of the parametric or the nonparametric modeling method being used, the skin and non-skin color models are calculated from the overall training samples. In other words, the skin and non-skin color distributions are global statistics. When applying them to skin detection in images or videos, the accuracy highly depends on the illumination of the images or videos. If the image or video is well illuminated, its skin color matches well with the global statistics and the performance will be satisfactory. On the contrary, if the illumination is unfavorable, the skin detection accuracy will degrade because the skin color distribution differs from the

* Tel.: +886 3 3412500x6071; fax: +886 3 3412173.
E-mail address: sunhm@mail.knu.edu.tw

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.09.022

general cases. To handle the possible change of the skin color model caused by the varying lighting conditions, dynamic adaptation techniques should be used.

Yang et al. [2] proposed an adaptive skin-color model for tracking the human face in videos. They use a Q–Q plot method [20] to verify that a mixture of multivariate Gaussian distributions can characterize skin color distribution. During the face tracking process, the skin color distribution of the current frame is considered a linear combination of the Gaussian distributions of the previous frame with a set of weighting factors, determined using a maximum likelihood criterion. Mckenna et al. [21] used a Gaussian mixture model to represent face color distributions in the HS space. The model evolves by joining the color probabilities calculated from the current frame; at frame t, the mean and standard deviation derive from the mean and standard deviation of frame $t-1$ by joining the color probabilities within a rectangular search area. The approach proposed by Sigal et al. [22] dynamically updates the skin color histograms based on feedback from the current segmentation and the prediction of a second-order Markov model. The evolution of the skin color histogram is parameterized by translation, scaling and rotation. In the system of Oliver et al. [23], skin color distribution is acquired from segmented blobs (i.e. the coherent regions containing homogeneous feature vectors) and modeled as a mixture of Gaussian distributions. An incremental Expectation–Maximization (EM) algorithm [24] continuously updates the parameters of the Gaussian mixture model during tracking.

The above methods adapt skin color models based on the assumption that there is a close relationship between contiguous video frames. However, no correlation exists between single images. The only information to utilize is in the image itself. Bergasa et al. [25] proposed an unsupervised and adaptive method for skin detection, which starts with pixel clustering. To simplify the classification procedure, some assumptions are made and, based on the assumptions, the classification can be completed by a set of discrimination functions, consisting of only the mean vector of each cluster. Chahir et al. [26] used an approach similar to the one proposed by Bergasa et al. The pixels in an image are clustered into several groups by a modified fuzzy c-means algorithm based on Hue Saturation Value (HSV) color values and the spectral distribution [27]. After the clustering process, the skin groups are identified. The above two approaches both start with pixel clustering and require that the skin occupies a relatively large portion in the image to make sure the skin colors can be successfully grouped together. Lee et al. [28] used a merging/grouping algorithm to derive five skin chroma clusters from the skin color present in a collection of naked images and then yielded five skin chroma distributions. For an unknown input image, five predefined areas around the image center are scanned to extract color features that are then fed into five pre-trained multilayer feed-forward neural networks to determine which skin chroma distribution is suited to this image for skin detection. This method relies heavily on color features collected from the five predefined areas. If no skin or less skin exists in these areas, the skin chroma will not be selected properly.

This paper presents a new approach for detecting skin in single images, which can conform to varying skin colors caused by different lighting conditions or races. First, a skin and a non-skin color model are built from the training images, called trained color models. The trained color models are used to select the pixels most likely to be skin in an unknown image. Then, the selected skin pixels are used to build a new skin color model specific to this image and this new model is combined with the trained skin color model to detect all the skin pixels. The underlying idea behind the proposed approach is that the trained skin model is tuned by imposing a local skin model to make the resulting skin model adapt to the current image. The experimental results are compared with Jones et al.'s. method based on the same testing database to show the effect of the new approach.

The remaining part of this paper is organized as follows. Section 2 describes the details of the proposed approach. Section 3 presents the parameter tuning tests and the experimental results. Some improvements for speeding up processing are also discussed and tested. Finally, Section 4 states some conclusions.

2. Proposed approach

The proposed approach consists of a training stage and a skin detection stage. In the training stage, a skin image data set and a non-skin image data set are scanned independently to calculate a skin color histogram and a non-skin color histogram (also referred to as the skin color model and the non-skin color model, respectively). These two histograms will be used in the skin detection stage. The skin detection stage includes three main steps: preliminary skin pixel detection, dynamic skin color modeling, and skin detection with the adapted skin color model. Fig. 1 shows the flowchart of the proposed approach. The following subsections discuss the details of each step.

2.1. Preliminary skin pixel detection

This step collects representative skin samples from the image to estimate the skin color distribution model specific to this image. To achieve this, we have to know what pixels are most likely to be skin. The trained skin color model and non-skin color model can be utilized to answer this question. It has been mentioned in Section 1 that the skin color model and non-skin color model can be viewed as the likelihood functions of the skin class and non-skin class, respectively. Thus, for a pixel p, its skin likelihood L_{skin} and non-skin likelihood $L_{non\text{-}skin}$ can be obtained by looking into the two models. Then, the likelihood ratio r is calculated by

$$r = \frac{L_{skin}}{L_{non\text{-}skin}}. \quad (1)$$

The larger the likelihood ratio r, the more likely that the pixel is skin. For each image, an accumulative histogram h can be made

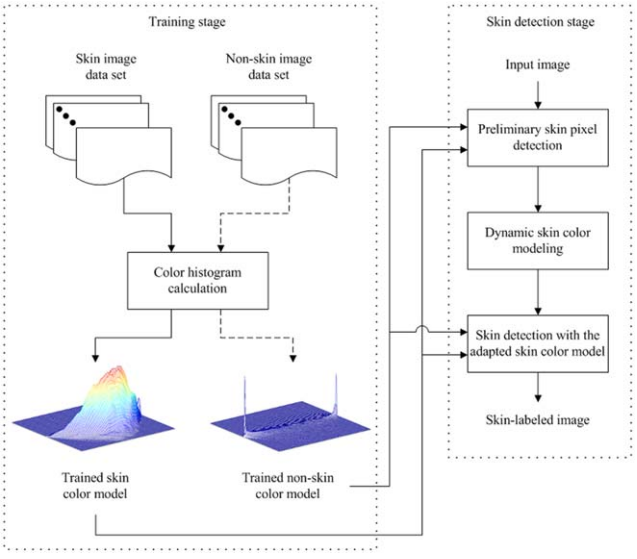

Fig. 1. Flowchart of the proposed approach.

as follows:

$$h(i) = |\{p | r_p \geq i\}|, i = 0, 1, 2, \ldots, \max \quad (2)$$

where r_p is the likelihood ratio of pixel p and max is the possible maximum of r. The accumulative histogram h records the number of pixels whose r_p quantities are larger than or equal to i. We can normalize the accumulative histogram h by dividing all its elements with $h(0)$ ($h(0)$ corresponds to the total number of pixels in the image). Fig. 2 shows an example of the normalized accumulative histogram. The closer to the max that the r_p quantity is, the more likely that p is skin. Thus, if we want to select a portion f of pixels, which are most likely to be skin, those pixels with $r_p \geq k$ must be selected, where k is the associated value of f in the normalized accumulative histogram, as shown in Fig. 2(b).

However, the images containing no skin may have pixels with relatively smaller r_p values. This means no good skin samples exist in the image. To deal with such cases, an additional threshold g is added and the preliminary skin pixel detection step is as below. Suppose we want to pick up a portion f of skin pixels from the image.

1. Examine the normalized accumulative histogram h to find the k value (i.e. the corresponding r_p value of f).

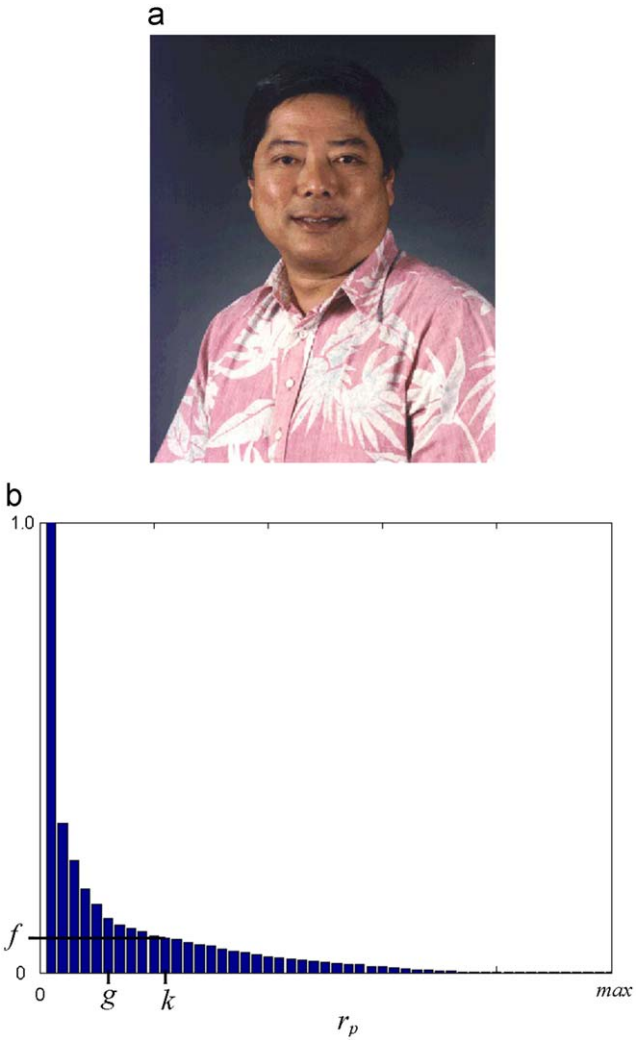

Fig. 2. (a) Image contains skin and (b) normalized accumulative histogram ($k > g$ in this case).

2. If $k \geq g$, select the pixels with $r_p \geq k$; otherwise, select only the pixels with $r_p \geq g$.

This step chooses pixels which are very likely to be skin based on the globally trained skin and non-skin color models. An issue, however, is how many pixels should be chosen. Selecting more pixels may help to make a good estimation of the color distribution of skin but non-skin pixels could be erroneously included too. Selecting fewer pixels can depress the probability of including non-skin pixels, however, the estimated skin color distribution could not match well with the real situation. This is a typical trade-off issue, controlled by parameters f and g. The proper settings of parameters f and g can be determined by a series of systematic tests and the details are discussed in Section 3.

2.2. Dynamic skin color modeling

Based on the collected skin pixels, the skin color distribution can be calculated for the image. It should be noted that these collected skin pixels are not random samples of all the skin pixels in the image, but a filtered output based on examining the likelihood ratio r. Therefore, the color distribution of the collected skin pixels may not be identical to the real skin color distribution in the image. Despite this, the color distribution of these collected skin pixels provides a useful cue to discover the range in which the skin pixels exist. Fig. 3 shows two skin images and their green–magenta-axis perspectives (i.e. projecting the 3-D color space along the green–magenta axis to a 2-D plane) of the collected skin pixels. The skin image in Fig. 3(a) contains bright skin color and the collected skin pixels gather on the right side of the histogram perspective as shown in Fig. 3(c). In contrast, Fig. 3(b) contains dark skin color and Fig. 3(d) shows that the collected skin pixels gather on the left side of the histogram perspective. With this knowledge, we can adapt the globally trained skin color model to fit the characteristic of the skin distribution of the image to improve the skin detection accuracy.

Given a set of data samples, the EM algorithm is a popular method for resolving the data distribution model through a mixture of several Gaussian probability density functions. The EM algorithm iteratively executes an expectation step, denoted as the E-step, and a maximization step, denoted as the M-step, to maximize the expectation of a specified likelihood function based on the data samples. The EM algorithm is also used in some skin detection systems which adopt Gaussian mixture models to represent the skin and non-skin color models [1,2,4,17,21,23]. Because of its computational cost, the EM algorithm is mainly used in the offline training stage. In our approach, constructing the skin model from the selected skin pixels (denoted as the local skin model in contrast with the globally trained skin model) must be completed online, hence a faster algorithm is needed. Also, the number of probability density functions involved in the Gaussian mixture model may vary according to different images, hence the algorithm must be able to determine it automatically for individual images. To meet the above requirements, a two-stage algorithm is used, which combines a maximin distance check and the K-means clustering algorithm [29]. The maximin distance check is responsible for determining the number of clusters (corresponding to the number of probability density functions) for the skin pixels and then the K-means algorithm completes the clustering task according to the cluster number. The following formally describes this scheme.

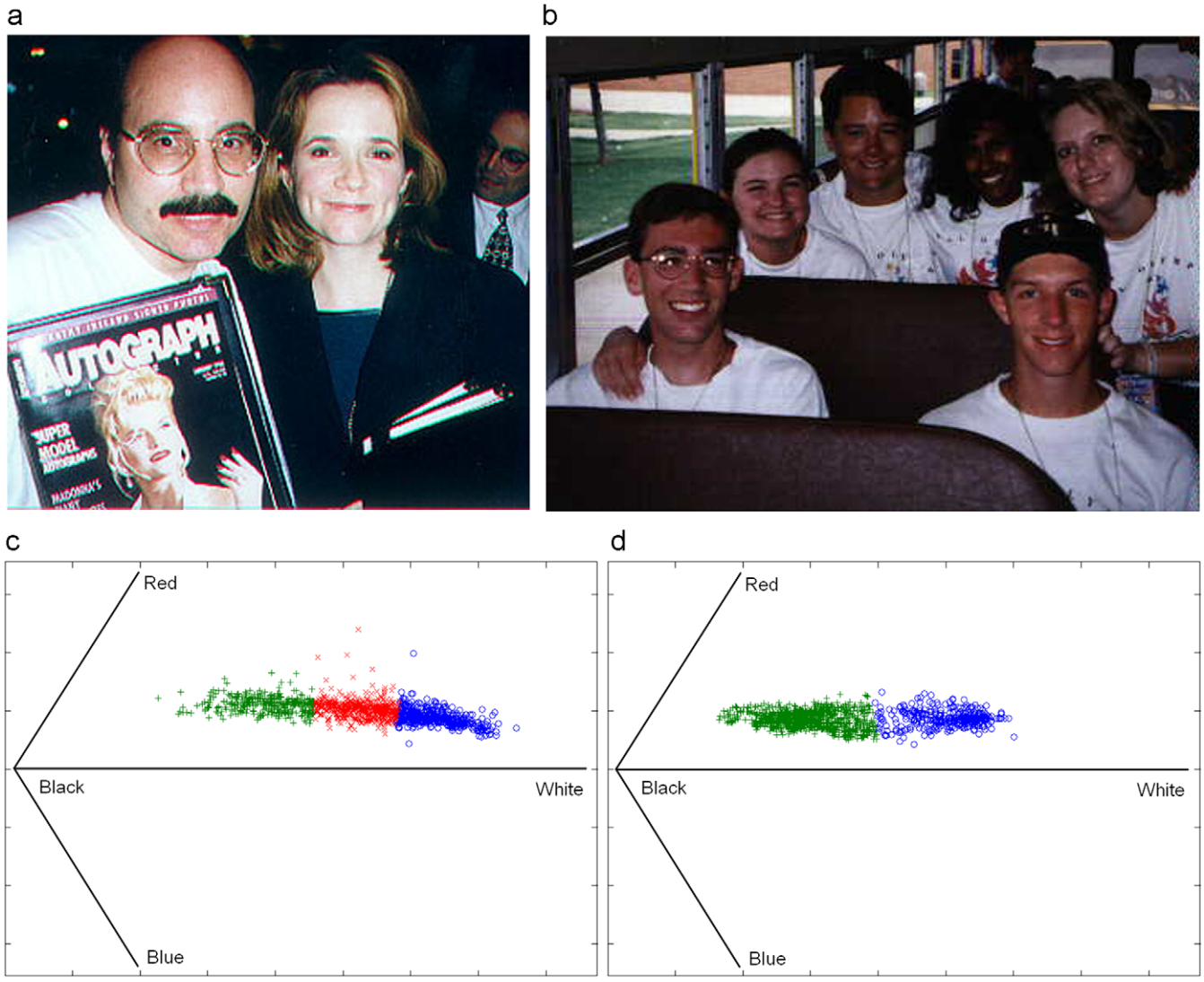

Fig. 3. (a) Image contains bright skin color, (b) image contains dark skin color, (c) skin pixels collected from the image of (a) (green-magenta-axis perspective view; the different colors represent the clustering result after applying our clustering algorithm), (d) skin pixels collected from the image of (b) (green-magenta-axis perspective view; the different colors represent the clustering result after applying our clustering algorithm). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Definitions.

(a) p_i, $i=1, 2,\ldots$, is the ith skin pixel;
(b) z_j, $j=1, 2,\ldots$, is the center of the jth cluster;
(c) ColorDist(p_i, z_j) is the Euclidean distance of the color of p_i and z_j in the color space;
(d) T is a threshold value for examining the color distance.

Algorithm. *Stage one*:

1. Set $z_1=p_1$.
2. For every p_i, calculate $D_i=\min_j\{\text{ColorDist}(p_i, z_j)\}$, where z_j, $1\le j\le n$, is the cluster center that has been determined.
3. Suppose $D_k=\max_i\{D_i\}$. If $D_k > T$, then set $z_{n+1}=p_k$ and go back to step 2; otherwise, terminate.

Stage two:

1. Classify the skin pixels using the minimum distance classification scheme (that is, assign p_i to cluster j, if ColorDist(p_i, z_j) $\le$ ColorDist(p_i, z_k), for all $k\neq j$).
2. Reset z_j to be the mean color of the skin pixels belonging to cluster j. If the value of each z_j does not change during two continuous iterations or the difference is less than a predetermined small threshold value, the algorithm terminates; otherwise, go back to step 1 for the next iteration.

The parameter T acts to control the number of resultant clusters; a larger T value leads to fewer clusters while a smaller T value results in more clusters. The T value is set empirically to be 80 in the present system. Figs. 3(c) and (d) show the result of applying the above clustering algorithm to the collected skin pixels. If we treat the pixels in each cluster as a normal distribution, the mean vector and the covariance matrix of each cluster can be computed. Then, the local skin color model of the image can be viewed as a linear combination of the individual normal distributions, and thus it can be calculated by the following formula:

$$P_{local}(p)=\sum_{i=1}^{N} w_i \frac{1}{(2\pi)^{3/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{(p-\mu_i)^T\Sigma_i^{-1}(p-\mu_i)}{2}\right) \quad (3)$$

where N is the number of clusters, μ_i and Σ_i are the mean vector and covariance matrix of cluster i, and w_i is the weighting factor associated with cluster i, which is calculated by

$$w_i = \frac{\text{Number of skin pixels belonging to cluster } i}{\text{Total number of skin pixels}} \quad (4)$$

Figs. 4(a) and (b) show the skin color models computed from the clusters of Figs. 3(c) and (d), respectively.

2.3. Skin detection with the adapted skin color model

The local skin color model acquired above provides an insight into the region in which the skin pixels exist in the color space for the image. However, as mentioned in Section 2.2, the local skin color model may not be identical to the real skin color distribution of the image because it derives from a filtered output, and not from random sampling. Thus, it cannot be used as the only guide for searching skin pixels in the image; instead, it may be used to adapt the trained skin color model to make the resultant model conform to the image characteristics. To this end, linearly combining the trained skin color model and the local skin color model with proper weighting factors could produce a new adapted skin color model. This could be done by the following formula:

$$P_{adapted}(p) = u_1 P_{trained}(p) + u_2 P_{local}(p) \quad (5)$$

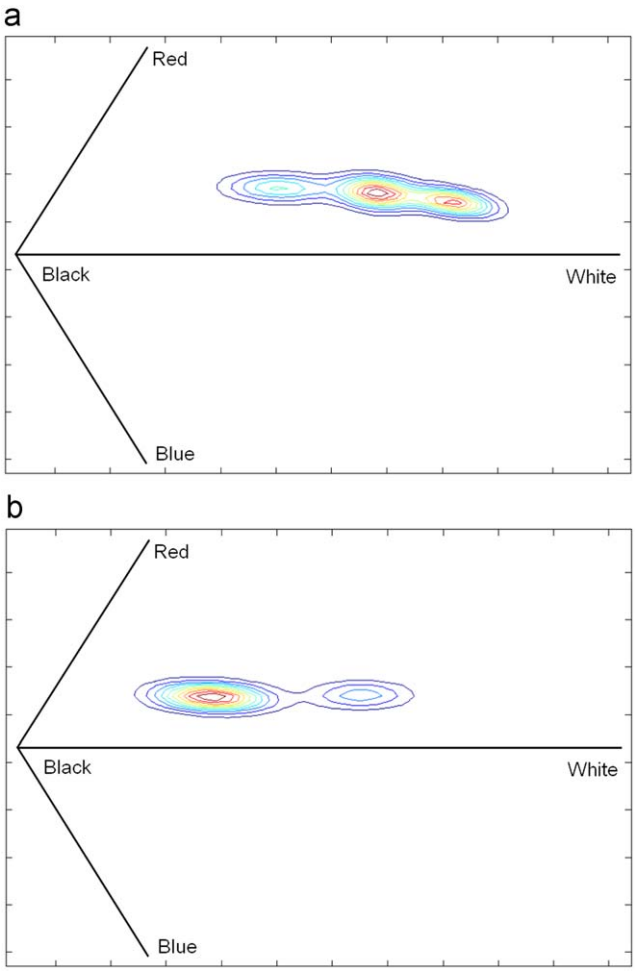

Fig. 4. Skin color models (green-magenta-axis perspective view): (a) computed from Fig. 3(c) and (b) computed from Fig. 3(d). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where the weighting factors u_1 and u_2 must satisfy the following conditions:

$$u_1 + u_2 = 1, \quad (6)$$

$$0 \le u_1 \le 1, \quad (7)$$

$$0 \le u_2 \le 1. \quad (8)$$

The proper values for u_1 and u_2 could be determined by experiments and the issue will be discussed in Section 3. Fig. 5 illustrates the original trained skin color model and the adapted skin color models after combining the trained model with the local skin models of Figs. 4(a) and (b) with u_1=0.8 and u_2=0.2. It can be seen that the adapted skin color models shift toward the color regions where the skin pixels exist in the image (bright side in Fig. 5(b) and dark side in Fig. 5(c)).

Based on the adapted skin color model, the adapted skin likelihood $L_{adapted}$ of a pixel p can be obtained; meanwhile, the non-skin likelihood $L_{non\text{-}skin}$ of pixel p can be found by looking into the trained non-skin color model. Then, based on the Bayes classification rule, the likelihood ratio of $L_{adapted}$ to $L_{non\text{-}skin}$ can be used to classify pixel p to be skin or not; that is

$$\frac{L_{adapted}}{L_{non\text{-}skin}} \ge \theta \quad (9)$$

where θ is a predetermined threshold, which determines the recognition behavior of the skin detection system. A larger θ value requires stronger evidence for a pixel to be skin and less skin pixels will be detected; as a result, the false detection rate (i.e. non-skin pixels are mistaken for skin pixels) can be reduced. On the contrary, a smaller θ value loosens the evidence for a pixel to be skin and more skin pixels will be found; as a result, the false detection rate can increase. This phenomenon can be depicted graphically using a receiver operating characteristic (ROC) curve and will be discussed in Section 3.

3. Experimental results

The proposed approach is not especially devoted to a certain color space. However, implementing a skin detection technique in different color spaces may cause a little performance difference in both accuracy and execution speed [4,19]. As the proposed approach will be compared with the original method proposed by Jones et al., which uses the RGB color space, we also implement the approach in the RGB color space.

Our experimental database is the same as that used by Jones et al., consisting of 8965 non-skin images and 4675 skin images. The skin areas in the skin images are labeled using outside mask files. We randomly select half of the non-skin images and half of the skin images to organize a training data set; the remaining images are organized as a testing data set. The training data set is used to build the trained skin color model and the trained non-skin color model.

As discussed in Section 2, three essential parameters f, g, and u_1 (u_2 is correlated to u_1 by $u_1+u_2=1$) may influence the performance of the proposed approach. To find out the proper settings for these parameters, we first empirically determine two values for each of the parameters: f=(0.05, 0.1), g=(1.2, 2.0), and u_1=(0.6, 0.8), and test all possible combinations of them (i.e. eight sets). Fig. 6 shows the ROC curves of the eight settings and the ROC curve of Jones et al.'s. method. It can be seen that the parameter setting (f, g, u_1)=(0.1, 1.2, 0.8) achieves the best accuracy in this test. Compared with the original method, the proposed approach significantly improves the skin recall rate (SRR) within the range of low false detection rates (FDRs). For example, on the ROC curve of (0.1, 1.2, 0.8), SRR is about 0.74 at FDR=0.05 while SRR is about 0.69 at the

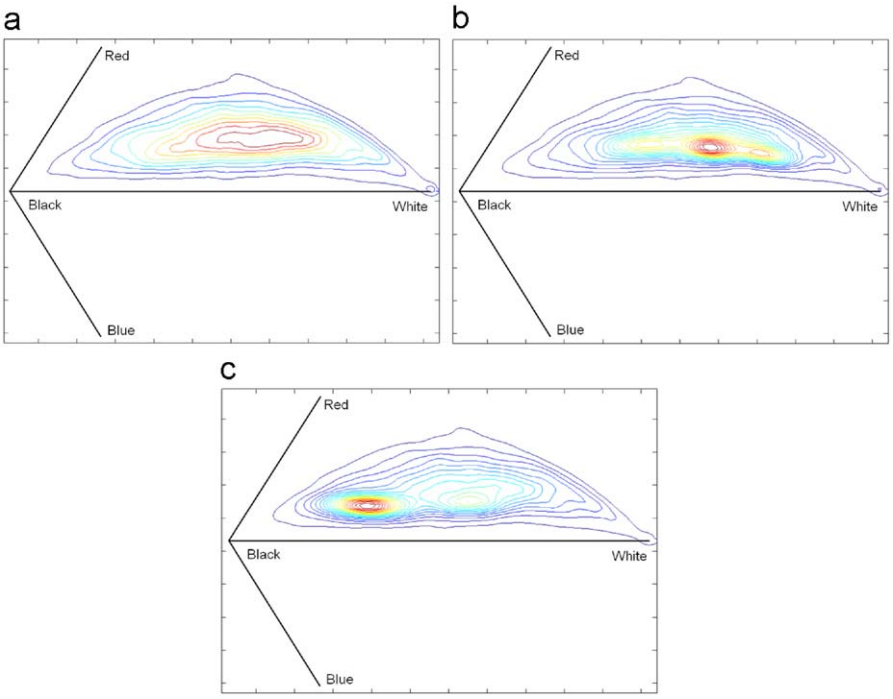

Fig. 5. (a) Trained skin color model. (b) Adapted skin color model obtained by combining the trained skin color model with the local skin color model of Fig. 4(a). (c) Adapted skin color model obtained by combining the trained skin color model with the local skin color model of Fig. 4(b). All the models are illustrated with the green-magenta-axis perspective view. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

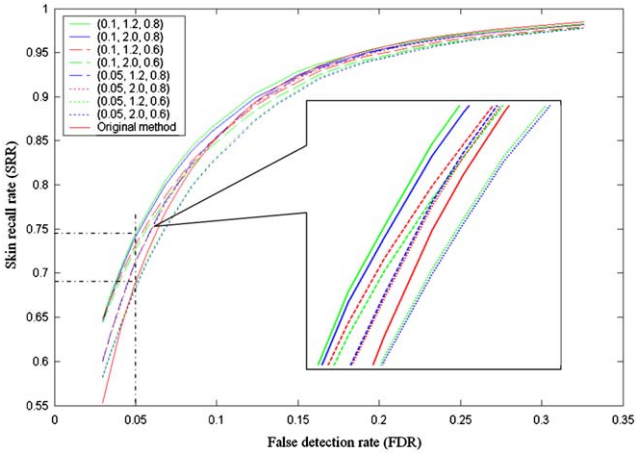

Fig. 6. ROC curves of the proposed method and Jones et al.'s method. The triple vectors in the legend correspond to the values of parameters f, g, and u_1 in order.

same FDR for the original method. However, in the range of high FDRs, such an advantage gradually decreases.

Fig. 7 illustrates two examples to visualize the difference between the proposed method and the original method (using the same threshold value θ=2.5). Fig. 7(a) shows the detected skin pixels (labeled as black) by the original method and Fig. 7(b) is the result of the proposed method (the source image is the one shown in Fig. 3(a)). In the image of Fig. 3(a), the skin region on the woman's face is overexposed and too bright compared with the normal skin color. The original method has a lower recognition rate for such a case, as shown in Fig. 7(a). By contrast, the proposed method can adapt to such variation and produce a better recognition result, as shown in Fig. 7(b). Figs. 7(c) and (d) are the processing results of the image in Fig. 3(b), in which the skin regions are underexposed. As illustrated, the proposed method also adapts to such cases and yields a better result than the original method.

A careful observation of the performance ranking of the different settings shows that f=0.1 is better than f=0.05 and also g=1.2 is better than g=2.0 and u_1=0.8 is better than u_1=0.6. This fact implies that superior results may be obtained by using larger f values, smaller g values, and larger u_1 values. Thus, we tried three more settings with (f, g, u_1) equal to (0.2, 1.0, 0.85), (0.25, 0.9, 0.9) and (0.3, 0.8, 0.95). Fig. 8 shows the results, together with the best setting in Fig. 6. It can be seen that the accuracy of the new settings is better than the previous settings. Also, based on the inspection, we could deduce that the optimal performance of the proposed approach may be achieved by (f, g, u_1)=(0.25, 0.9, 0.9).

However, the computational cost of the proposed approach is more than the original method because the image must be scanned to pick up skin samples. The clustering process and local skin model generation also require additional calculation. On average, the original method takes 0.073 s and our approach takes 0.313 s for each image (including file-reading time). Compared with the original method, the proposed approach takes more processing time and may cause undesirable process delay. However, two improvements can increase the processing speed. First, downsampling selected skin pixels can reduce the skin pixel number to speed up the clustering process. Fig. 9 shows the result of testing different downsampling factors. Table 1 tabulates the processing time associated with the different downsampling factors. The experimental results show that the downsampling factors <64 do not affect the accuracy and may save up to 43.8% of the processing time.

The second improvement is to use the exp mathematical function supplied by the Intel Integrated Performance Primitives (IPP) function library. The IPP library is developed by Intel company to fully utilize advanced CPU computation features (e.g. SSE2 instruction set) to speed up calculation. The exp function is naturally time consuming and heavily invoked in the local skin model generation. Based on our test, replacing the

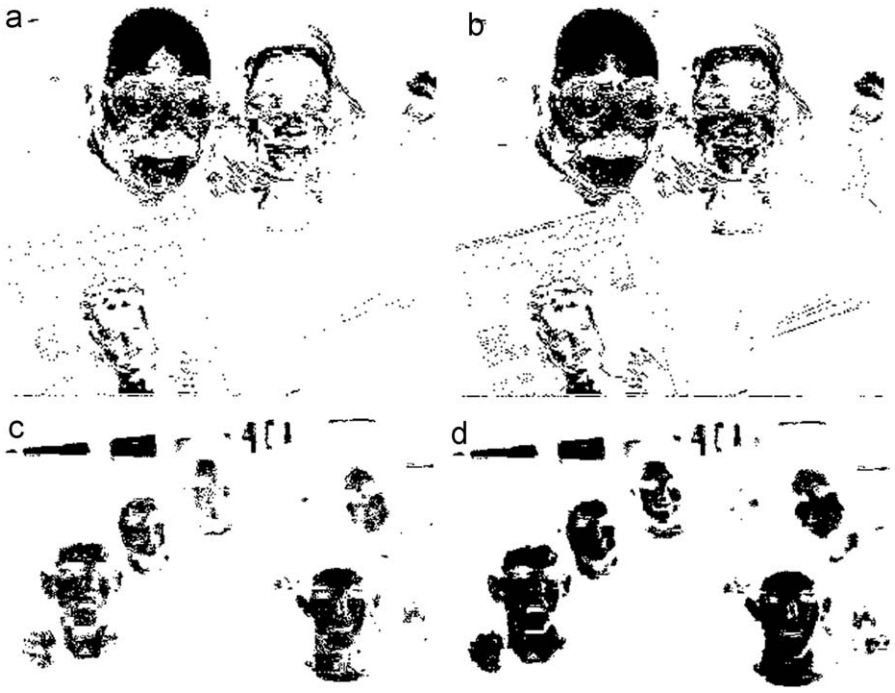

Fig. 7. (a) and (b) are the skin pixels detected by the original method and the proposed method, respectively (the source image is shown in Fig. 3(a)). (c) and (d) are the skin pixels detected by the original method and the proposed method, respectively (the source image is shown in Fig. 3(b)).

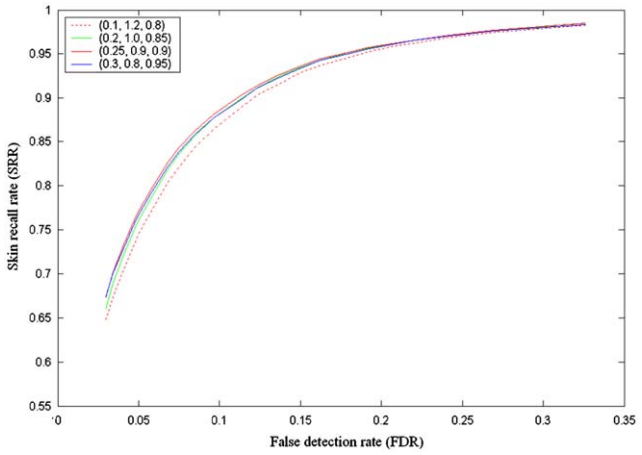

Fig. 8. ROC curves of (f, g, u_1)=(0.1, 1.2, 0.8), (0.2, 1.0, 0.85), (0.25, 0.9, 0.9) and (0.3, 0.8, 0.95).

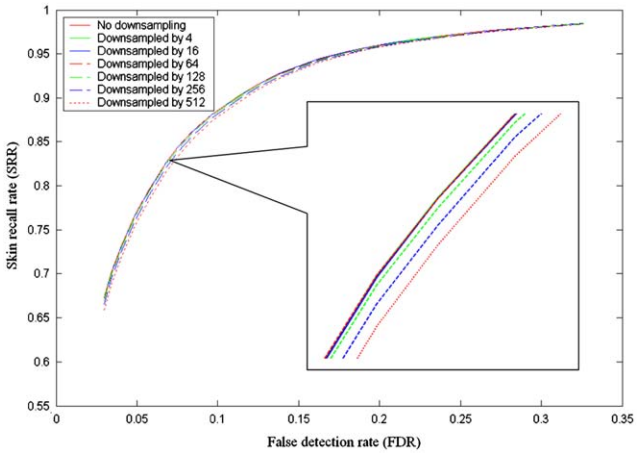

Fig. 9. Experimental results of applying different downsampling factors to the selected skin pixels. Downsampling factor s means randomly selecting one skin pixel from per s skin pixels.

generic exp function with the IPP exp function saves 18% of the processing time. Applying the two improvements reduces the processing time for each image to 0.144 s.

4. Conclusions

This paper proposes a new skin detection approach for detecting skin in a single image. Earlier skin detection techniques suffer from accuracy degradation caused by varying illumination conditions in images or videos. To improve this defect, several adaptive skin color modeling methods have been proposed for tracking skin regions in videos, which utilize the correlation between contiguous frames. For stand-alone images, no correlation exists between them and thus a new method is needed. Our approach uses a local skin color model to shift the globally trained skin color model to adapt the final skin model to the current image. The local skin color model is generated from skin pixel samples collected from the image. The performance of this approach depends on the number of selected skin pixel samples

Table 1
Processing time of different downsampling factors.

Downsampling factor	Average processing time per image (s)	Reduced time compared with no downsampling (%)
4	0.226	27.8
16	0.192	38.7
64	0.176	43.8
128	0.163	47.9
256	0.151	51.8
512	0.138	55.9

and the weighting factors for combining the local skin color model and the trained skin color model. A series of experiments are conducted to find the optimal parameter settings. With the optimal parameter settings, the approach achieves better accuracy at low false detection rates compared with the original method. At high false detection rates, its accuracy is almost the same as the original method.

Generating a local skin color model requires more computation than the original method. To speed up processing, two improvements are made by downsampling skin pixel samples and using IPP exp function. Our test shows that the improvements save 54% of the processing time without decreasing the accuracy.

References

[1] M.J. Jones, J.M. Rehg, Statistical color models with application to skin detection, International Journal of Computer Vision 46 (1) (2002) 81–96.

[2] J. Yang, W. Lu, A. Waibel, Skin-color modeling and adaptation, in: Proceedings of the ACCV98, 1998, pp. 687–694.

[3] M. Storring, H.J. Andersen, E. Granum, Physics-based modelling of human skin colour under mixed illuminants, Robotics and Autonomous Systems 35 (3) (2001) 131–142.

[4] P. Kakumanu, S. Makrogiannis, N. Bourbakis, A survey of skin-color modeling and detection methods, Pattern Recognition 40 (3) (2007) 1106–1122.

[5] D. Chai, K.N. Ngan, Face segmentation using skin-color map in videophone applications, IEEE Transactions on Circuits and Systems for Video Technology 9 (4) (1999) 551–564.

[6] Y. Wang, B. Yuan, A novel approach for human face detection from color images under complex background, Pattern Recognition 34 (10) (2001) 1983–1992.

[7] Y. Dai, Y. Nakano, Face-texture model based on SGLD and its application in face detection in a color scene, Pattern Recognition 29 (6) (1996) 1007–1017.

[8] M.J. Seow, D. Valaparla, V.K. Asari, Neural network based skin color model for face detection, in: Proceedings of the 32nd Workshop on Applied Imagery Pattern Recognition, 2003, pp. 141–145.

[9] S.L. Phung, D. Chai, A. Bouzerdoum, A universal and robust human skin color model using neural networks, in: Proceedings of the IJCNN01, 2001, pp. 2844–2849.

[10] C. Chen, S.-P. Chiang, Detection of human faces in colour images, IEE Proceedings of Vision, Image and Signal Processing 144 (6) (1997) 384–388.

[11] H. Zheng, M. Daoudi, B. Jedynak, Blocking adult images based on statistical skin detection, Electronic Letters on Computer Vision and Image Analysis 4 (2) (2004) 1–14.

[12] C.-H. Wu, P.C. Doerschuk, Tree approximations to Markov random fields, IEEE Transactions on PAMI 17 (4) (1995) 391–402.

[13] D. Chai, A. Bouzerdoum, A Bayesian approach to skin color classification in YCbCr color space, in: Proceedings of the IEEE TENCON00, vol. 2, 2000, pp. 421–424.

[14] T.-W. Yoo, I.-S. Oh, A fast algorithm for tracking human faces based on chromatic histograms, Pattern Recognition Letters 20 (10) (1999) 967–978.

[15] S. Srisuk, W. Kurutach, A new robust face detection in color images, in: Proceedings of the IEEE AFGR02, 2002, pp. 306–311.

[16] P. Kuchi, P. Gabbur, P.S. Bhat, S. David, Human face detection and tracking using skin color modeling and connected component operators, IETE Journal of Research 38 (2002) 289–293.

[17] M.-H. Yang N. Ahuja, Gaussian mixture model for human skin color and its applications in image and video databases, in: Proceedings of the SPIE Conference Storage and Retrieval for Image and Video Databases, 1999, pp. 458–466.

[18] H. Greenspan, J. Goldberger, I. Eshet, Mixture model for face-color modeling and segmentation, Pattern Recognition Letters 22 (14) (2001) 1525–1536.

[19] V Vezhnevets, V Sazonov, A Andreeva, A survey on pixel-based skin color detection techniques, in: Proceedings of the Graphicon, 2003, pp. 85–92.

[20] J.D. Jobson, Applied Multivariate Data Analysis: Regression and Experimental Design, vol. I, Springer, Berlin, 1991.

[21] S.J. Mckenna, S. Gong, Y. Raja, Modelling facial colour and identity with Gaussian mixtures, Pattern Recognition 31 (12) (1998) 1883–1892.

[22] L. Sigal, S. Sclaroff, V. Athitsos, Skin color-based video segmentation under time-varying illumination, IEEE Transactions on PAMI 26 (7) (2004) 862–877.

[23] N. Oliver, A.P. Pentland, F. Berard, LAFTER: lips and face real time tracker, in: Proceedings of the Computer Vision and Pattern Recognition, 1997, pp. 123–129.

[24] M. Redner, H. Walker, Mixture densities maximum likelihood and the em algorithm, SIAM Review 26 (1984) 195–239.

[25] L.M. Bergasa, M. Mazo, A. Gardel, M.A. Sotelo, L. Boquete, Unsupervised and adaptive Gaussian skin-color model, Image and Vision Computing 18 (12) (2000) 987–1003.

[26] Y. Chahir, A. Elmoataz, Skin-color detection using fuzzy clustering, in: Proceedings of the ISCCSP, 2006.

[27] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (1973) 32–57.

[28] J.-S. Lee, Y.-M. Kuo, P.-C. Chung, E.-L. Chen, Naked image detection based on adaptive and extensible skin color model, Pattern Recognition 40 (8) (2007) 2261–2270.

[29] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974.

About the Author—HUNG-MING SUN received the PhD degree in Computer Science from the National Cheng Kung University, Taiwan. He was a senior researcher in Ulead Systems, Inc. He is currently an assistant professor in the Department of Computer Science and Information Engineering, Kainan University, Taiwan. His research interests include pattern recognition, image analysis, MPEG-7, and UML.