Multimedia Systems (2009) 15:309–323
DOI 10.1007/s00530-009-0165-1

REGULAR PAPER

A new dynamic cellular learning automata-based skin detector

Ahmad Ali Abin · Mehran Fotouhi · Shohreh Kasaei

Published online: 15 September 2009

Abstract Skin detection is a difficult and primary task in many image processing applications. Because of the diversity of various image processing tasks, there exists no optimum method that can perform properly for all applications. In this paper, we have proposed a novel skin detection algorithm that combines color and texture information of skin with cellular learning automata to detect skin-like regions in color images. Skin color regions are first detected, by using a committee structure, from among several explicit boundary skin models. Detected skin-color regions are then fed to a texture analyzer which extracts texture features via their color statistical properties and maps them to a skin probability map. This map is then used by cellular learning automata to adaptively make a decision on skin regions. Conducted experiments show that the proposed algorithm achieves the true positive rate of about 86.3% and the false positive rate of about 9.2% on Compaq skin database which shows its efficiency.

Keywords Skin detection · Skin map · Skin color space · Texture analysis · Cellular learning automata

This work was in part supported by a grant from *Iran telecommunication research center* (ITRC). We also would like to thank Dr. Beigy for his help on the theory and implementation of CLA.

A. A. Abin · M. Fotouhi · S. Kasaei (✉)
Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
e-mail: skasaei@sharif.edu

A. A. Abin
e-mail: abin@ce.sharif.edu

M. Fotouhi
e-mail: fotouhi@ce.sharif.edu

1 Introduction

Skin detection refers to isolation of image (or video) pixels that correspond to human skin. It is very useful for detecting human body parts (faces, hands, etc.) in numerous applications, such as face detection [1], face tracking [2], filtering of objectionable web images [3], and human-computer interaction [4,5] to mention a few. While multimedia data are becoming popular in wireless ad hoc environments, some methods try to cluster such multimedia data based on their semantic contents [6]. Skin color can be an appropriate option for this type of semantic clustering. More important properties of *skin detection* (SD) include increasing the total processing speed (as only skin regions will be processed) and being invariant against rotation, scale, partial occlusion, and pose change. However, existence of various ethnic and individual characteristics (e.g., race, age, body part) results in having different skin colors appearances. Figure 1 shows such different skin colors. As such, the main challenge is to make the SD process robust against large appearance variations that can occur. In fact, skin appearance is affected by changes in intensity, color, and emitting light direction. Camera characteristics, cast shadows of other objects within the scene, and additional light reflections also affect the skin appearance. Occlusions (clothing, hair, eye glasses, etc.) can also introduce major difficulties in SD. Furthermore, other objects might also exist in the scene with skin-like colors (e.g., certain types of wood, sand, clothes, and paintings). Figure 2 shows some different skin appearances in natural images.

When developing a system that employs skin color as a feature for SD, the researchers usually face two major questions: which color space should be chosen and how exactly the skin color distribution should be modeled. Some researchers investigate the effect of the chosen color space

Fig. 1 Different skin colors [7]

on detected skin regions [8,9] and the aim of some others is to model the skin color distribution [10,11]. Some authors define explicitly (through a number of rules) the skin cluster boundaries in some color spaces [12]. A good survey on the performance of different color spaces for SD and explanation of skin color modeling can be found in [12,13]. In Sect. 2.1 a review on some of these methods is given in detail.

In this work, we consider the *cellular learning automata* (CLA) as a model for systems that consist of simple elements. In fact, these simple elements (which have the learning capability) act jointly to represent the complicated behavioral patterns. A CLA is the cellular automata (CA) in which a *learning automata* (LA) is assigned to each cell. The learning automaton located in a particular cell determines its state (action) based on its action probability vector. The assigned rule of CLA and the actions selected by the neighboring LA of any particular learning automaton determine the reinforcement signal to the learning automaton located in a cell. These simple elements improve their performance based on the behavior of their neighbors and previous experiences. Nevertheless, they can expose complex behavior based on their interactions. The neighborhood properties among image pixels make CLA as a good candidate for image processing tasks. A number of applications of CLA have been developed recently: namely, image processing [14], cellular mobile networks [15], modeling of commerce networks [16], solving NP-Complete problems [17], capacity assignment [18], and neural networks [19].

In this paper, we propose an algorithm that combines color and texture information of skin regions with CLA to detect skin regions in color images more accurately. Skin color regions in the input image are first detected by using a committee structure to make a decision from several explicit boundary skin models [20]. This structure maintains the benefits of different color spaces. The detected skin-color regions are then fed to a texture feature extractor. Two approaches are employed to extract texture features of skin regions, via their statistical properties, which are then used to produce a skin probability map. The probability map is then fed to a cellular learning automaton to make the final decision on skin regions.

The remainder of this paper is organized as follows. Section 2 presents a review on related work on SD and summarizes some related concepts of CLA. Proposed SD algorithm is presented in Sect. 3. Experimental results are discussed in Sect. 4. Finally, concluding remarks are given in Sect. 5.

2 Related work

This section starts by presenting the basic structure of SD methods and presents a review on some related work on SD in Sect. 2.2. A summary of some related concepts of CLA and its structure is described in Sect. 2.2.

2.1 Skin detection

Skin detection aims at detecting human skin pixels in color images. The process results in a binary image (map) that indicates the location of obtained skin pixels (see Fig. 3). In SD process, a primary task is to choose a suitable color space to represent image pixels and their related skin color distribution model. Some color spaces are able to represent skin colors more compactly. For example, human skin color in Nrgb and YC_bC_r color spaces are more compactly represented in chromaticity space than in other color spaces, such as RGB and HIS [21,22]. Dia and Nakano [23] used the "I" component of YIQ color space to detect skin pixels from images that mostly contain people with yellow skins. A good assessment of different color spaces for skin color representation and comparative evaluations among them can be found in [22,24].

In general, some factors such as the chosen color space, the amount of training data, and the application requirements can affect the skin distribution modeling. In some applications,

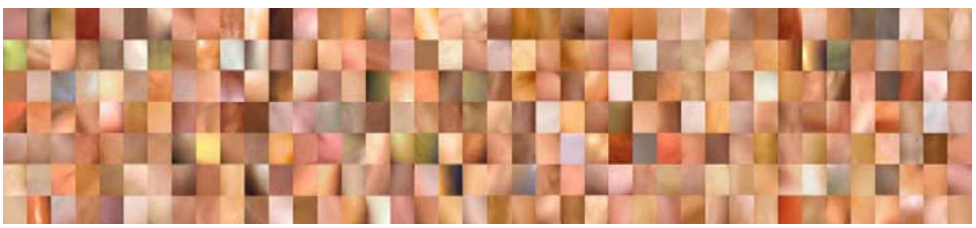

Fig. 2 Different skin appearances in natural images [7]

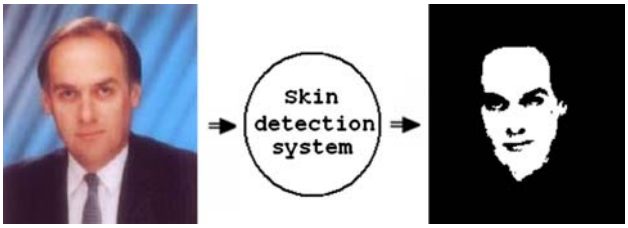

Fig. 3 A typical image of Compaq database and its corresponding skin mask

SD is used as a preprocessing step and thus a fast SD method is required. In some applications, hardware constraints demand the use of specific methods. And finally, in some applications, such as human tracking or face tracking, the change of environmental lighting affects the choice of color space and skin modeling. Note that SD in images with complex background and non-human objects requires more complicated and time-consuming methods.

Skin detection can also be considered as a classification method having two classes: skin and non-skin. SD app roaches that only use the pixel color are called pixel-based skin detectors. These approaches are based on the same pixel-wise processing paradigm, in which each image pixel is individually analyzed. Jones and Rehg [25] claimed that in most color spaces the skin colors distribute in some compact regions. One of the simplest methods of this kind is to define some rules to explicitly characterize the skin color boundaries in a color space under an illumination-controlled environment [26]. Explicitly defined skin color boundaries in different color spaces are frequently used in the literature [20,27–30] because of their simplicity and computational efficiency. Brown et al. [31] used the *self-organizing map* (SOM) to determine skin clusters. Several color spaces have been tested for SOM, but the conducted experiments have shown that the performance of SOM-based SD methods is not affected by the used color space.

Another simple SD method is the histogram modeling [32–34], or the normalized *lookup table* (LUT). In this method, a 2-D, or 3-D color histogram is used to model the skin distribution in different color spaces. After the training stage, the histogram counts are normalized to convert the histogram values to a discrete probability distribution. In histogram modeling, the conditional probability histogram of the observing color, c, given skin, $P(c|skin)$, and non-skin, $P(c| \sim skin)$ data is computed (using the Bayes rule). The obtained probability is then compared with a predetermined threshold to obtain the SD rule [25,35,36]. The *receiver operating characteristics* (ROC) curve is usually computed to indicate the true positive rate versus the false positive rate for different thresholds to evaluate the detection performance. Both of these methods need to use a large database for training and enough memory to hold the large lookup table. To overcome these drawbacks, the parametric skin distribution modeling is used.

Under controlled illumination conditions and for similar skin color types, an elliptical Gaussian joint probability (single Gaussian model) is used to model skin color distribution in [1,36,37]. Parameters of the model can be achieved by parameter estimation approaches from training data (such as the maximum likelihood). The obtained probability is then compared with a certain threshold to measure the skin color likeliness. Alternatively, the Mahalanobis distance (λ) can be used to classify the pixel color as skin or non-skin [38,39] pixel color.

Yang and Ahuja [40] used a statistical test to show that single Gaussian model is not adequate to model the skin distribution in LUV color space for the considered dataset. A more advanced and complicated model to describe complex shaped distributions is the *Gaussian mixture model* (GMM), the generalization of the single Gaussian model. Skin color types and varying illumination conditions can be modeled using GMM, where the number of Gaussian components is assumed to be known beforehand. Parameters of this model can be obtained by *expectation maximization* (EM) algorithm, where it needs proper initial conditions to converge. The details of training GMM with EM can be found in [38,40]. Choice of the number of Gaussian components is based on the used color space and the number of skin types in the training data. Researchers have used 2 [40] to 16 [25] number of Gaussian components in their models. The effect of this choice is discussed in [39] for Nrgb color space. In [41], GMM has been used for skin color in a feature space created by a non-linear map defined empirically by the use of a polynomial estimated kernel.

Lee and Yoo [42] investigated the skin and non-skin distribution in several color spaces. They concluded that the shape of skin color cluster is almost elliptical. They claimed that the usage of a symmetric Gaussian model leads to a high false positive rate and hence proposed an elliptical boundary model. They compared their model with a single and mixture of Gaussian model on Compaq database. In [43], a maximum entropy classifier is used for estimating skin probability distribution from histogram of one pixel and its two adjacent pixels. Then, the Gibbs sampler is used to derive the probability of skin for each pixel. Other classifiers such as *multi-layer perceptron* (MLP) [44,45], *Bayesian network* (BN) [46], and *support vector machine* (SVM) [47] are also used for SD. Fuzzy approaches are also used for pixel-based SD [48]. A comparative assessment of traditional pixel-based SD methods can be found in [11,12]. However, standard SD techniques are not robust enough to deal with complex environments containing surfaces and objects with skin-like colors.

In order to detect skin regions from non-human skin-like objects, several region-based methods have been proposed,

which take into account the spatial relationship among image pixels. In fact, the texture features of skin-like regions are counted to better distinguish between skin and non-skin areas. As the pixel-based SD is the primary stage of these methods, the performance of the region-based SD still depends on the outcome of pixel-based SD process. The idea behind the region-based SD methods is that human beings can detect skin regions in real scenes without specific difficulties. However, for a human being the classification of a single pixel as skin or non-skin is a very difficult task. In fact, humans use many high-level processes to assist the detection of skin (detection of hair, clothes, etc). They also use some simple diffusion mechanisms to detect colors and textures.

Diplaros et al. [49] assumed that each pixel in an image has a hidden binary label associated with it that specifies whether it is a skin pixel or not. A variation of EM algorithm which incorporates the spatial constraints is then used to solve the inference problem. In [50], a shape and color-based model for SD has been proposed. The model is parametric and represents the spatial arrangement of skin pixels as compact elliptical regions. Its parameters are estimated by maximizing the mutual information between the distribution of model-generated skin pixels and the distribution of skin colors. Forsyth and Fleck [51] have subtracted the original image from its median filtered image to extract the texture of skin-like areas. In [52], the mean and standard deviation of the magnitude of Gabor wavelet coefficients are considered as texture features. Texture classification is then applied by using GMM.

In general, the robustness of SD decreases under varying lighting conditions. Therefore, one of the most important features of skin color model (especially in video applications) is its ability to adapt to the changes in lighting and viewing conditions. In order to tolerate varying illuminations, two major approaches exist. Illumination invariance (or color constancy) is the most common approach for dealing with lighting changes. In this approach, the image content is transformed to a known canonical illuminant to estimate the illuminant color. Then, the image is enhanced pixel-wise based on the estimate of illuminant [53–55]. Skin color modeling and detection are then applied on these color-enhanced images [56–58]. In this approach, the color constancy algorithm is applied as a preprocessing step. A good comparison of different approaches with respect to color constancy is reported in [59,60]. Alternatively, the adaptive (or dynamic) approaches are to change the previous developed color model into the new environment. Both pixel-based and region-based SD methods can be adapted for illumination changes. Two common methods for dynamical modeling of skin color distribution are histogram and GMM [57,61]. The advantage of histogram approach is that the probability distribution can be computed inexpensively. Also, if the number of Gaussian components is known in prior, the parameters of GMM can be updated in real-time. Adaptive thresholding is an intuitive dynamic approach to detect skin pixels in varying illuminations. For instance, Cho et al. [62] proposed a method based on adaptive explicit thresholding in HSV color space. In [12] and [24], a short review on illumination change handling approaches is given. Fuzzy approaches are also presented to overcome illumination changes in [63,64].

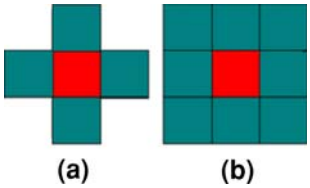

Fig. 4 Two neighborhood options: **a** Von Neumann, **b** Moore [14]

2.2 Cellular automata

Cellular automata were invented in the 1940s by the mathematicians John Von Neumann and Stanislaw Ulam [65]. Automata are simple agents for doing simple tasks by design. The popularity of CA comes from their simplicity and enormous potential in modeling complex systems. CA can be viewed as a simple model of a spatially extended decentralized system made from a number of individual components (cells) [66,67]. They consist of an array of cells, where each cell is allowed to be in one of a few states. Each cell considers its neighbors to know their states. Based on this information, each cell applies a simple rule to determine whether it has to alert its state [68]. The communication between constituent cells is limited to only local interactions. The overall structure can be viewed as a parallel processing task. However, this simple structure when iterated for several times produces complex patterns displaying the potential to simulate different sophisticated natural phenomena. Two possible neighborhood options are shown in Fig. 4.

The states of all cells in the lattice are described by a configuration that can be described as the state of the whole lattice. The rule and the initial configuration of CA specify the evolution of CA that indicates how each configuration is changed in one step. An approach to solve the learning problem was initiated by the Russian mathematician Tsetlin [69]. He introduced, in 1961, a new mode1 of computer learning that is now called learning automata. LA interact with a random environment to improve their behavior. The term environment refers, in general, to the collection of all external conditions and influences affecting the development of a system. LA are connected to the environment in a feedback loop, such that the input of that automata is the output of the environment, and the vice versa (see Fig. 5).

The functionality of LA can be described in terms of a sequence of repetitive feedback cycles in which the automata

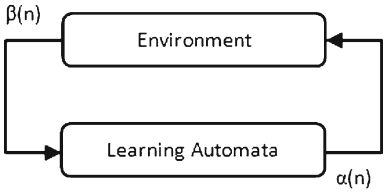

Fig. 5 Feedback connections of learning automata and their environment [70]

interact with an environment. The automata have a finite set of actions, α_i, to choose from, and at each stage, k, the choice of an action depends upon their action probability vector, $\rho_i(k)$. The automata choose an action that triggers a response (reinforcement signal) from the environment. Such a response can be either a reward or a penalty. The automata use this response and the knowledge acquired in previous actions to determine the next action. The automata update their probability vector of actions depending upon the reinforcement signal received at that stage. Various learning algorithms have been reported for LA. For example, in the linear reward-εpenalty, $L_{R-\varepsilon P}$, scheme the recurrence equation for updating ρ is defined by [70]

$$\rho_j(k+1) = \begin{cases} \rho_j(k) + \alpha \times (1 - \rho_j(k)) & \text{if } i = j \\ (1-\alpha) \times \rho_j(k) & \text{if } i \neq j \end{cases} \quad \text{for } \beta(k) = 0 \tag{1}$$

$$\rho_j(k+1) = \begin{cases} (1-b) \times \rho_j(k) & \text{if } i = j \\ \frac{b}{r-1} + (1-b) \times \rho_j(k) & \text{if } i \neq j \end{cases} \quad \text{for } \beta(k) = 1 \tag{2}$$

in which α_i denotes the action chosen at time k as a sample from probability distribution $\rho(k)$, and $\beta(k)$ is the reinforcement signal received from its environment.

If the environment rewards the selected action, then $\beta(k)=0$ or else $\beta(k)=1$. Parameters $0 < b < \alpha < 1$ represent the step lengths, and r is the number of actions for LA. Parameters a and b are the increment and decrement factors of the action probabilities. If $\alpha = b$, then Eqs. (1) and (2) are called the linear reward-penalty, L_{R-P}, algorithm, and if $b = 0$ they are called the linear reward-inaction, L_{R-I}, algorithm. Some applications of LA include telephone and traffic routing and control [71], stochastic geometric problems [72], stochastic point location problem [73], game theory, and pattern recognition [74].

2.3 Cellular learning automata

The full potential of LA is realized when multiple automata interact with each other. Interaction may assume different

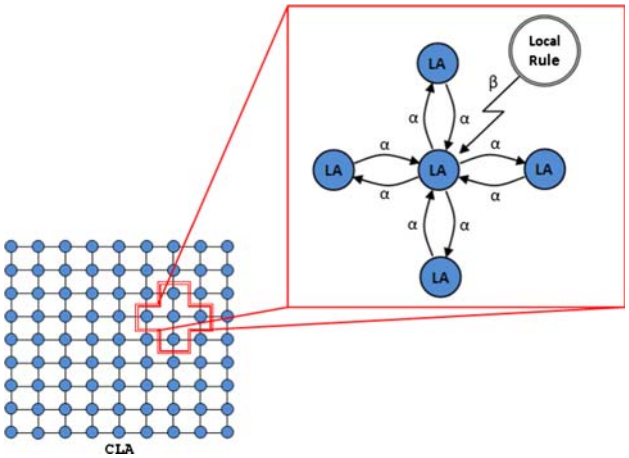

Fig. 6 Illustration of cellular learning automata [70]

forms (e.g., tree, mesh, array, etc). Depending on the problem under consideration, one of these structures for interaction may be chosen. In most applications, a full interaction between all LA is not necessary and natural. Local interactions of LA, which can be defined in the form of a graph (such as a tree, mesh, or array) are natural in many applications. On the other hand, CA are mathematical models for systems containing a large number of simple identical components with local interactions. If CA and LA combine together, a new model called cellular learning automata is created (see Fig. 6). This model is superior to CA because of its ability to learn. It is also superior to a single LA because it is a collection of LA which can interact with each other. The basic idea of CLA, which is a subclass of the stochastic CA, is the usage of LA to adjust the state transition probability of stochastic CA [70].

In general, as mentioned above, CLA form a mathematical model for dynamic and complex systems that consist of a large number of simple components. These simple components (which have the learning capability) act jointly to produce complicated behavioral patterns. CLA contain CA in which LA are assigned to every cell. The LA residing in a particular cell determines its state (action) based on their action probability vector. The underlying rule of CLA and the actions selected by the neighboring LA of any particular LA determine the reinforcement signal (rewarding or penalizing signal) to the LA located in a cell. The neighboring LA of any particular LA constitutes the local environment of that cell. The local environment of a cell is non-stationary because the action probability vectors of the neighboring LA vary during the evolution of CLA.

Overall iterative behavior of CLA is illustrated in Fig. 7 which can be summarized as follows:

- Determine the state of every cell based on action probability vectors of LA located in that cell.

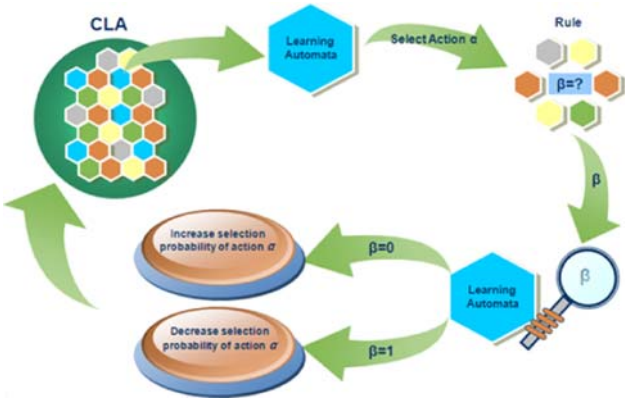

Fig. 7 Functionality of cellular learning automata

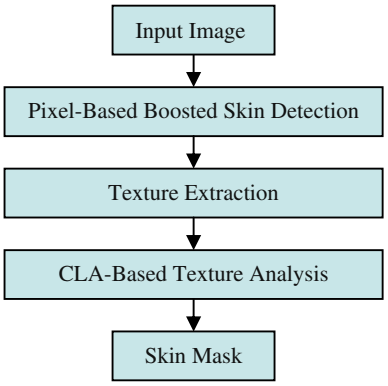

Fig. 8 Overall structure of proposed skin detection method

- Choose the initial values of this state based on previous experiences, or at random.
- Consider the CLA rule to determine the reinforcement signal to LA located in that cell.
- Update the action probability vector of LA based on the supplied reinforcement signal and the chosen action by that cell.
- Iterate the process until the desired result is obtained.

A number of applications for CLA have been developed recently. These include image processing [14,75,76], cellular mobile networks [15], modeling of commerce networks [16], solving NP-complete problems [17], capacity assignment [18], and neural networks [19]. In [70], a mathematical methodology to study the behavior of CLA is given and its convergence properties are investigated.

3 Proposed method

In this section, the proposed SD method that combines color and texture information of skin regions with cellular learning automata to detect skin areas in color images is presented. Figure 8 shows the overall structure of the proposed SD method. As shown in this figure, a pixel-based boosted SD is used to recognize skin-like pixels. The output of this part is fed to a texture analyzer which extracts texture features from the skin-like region. As the texture contains some information about the coarseness of the region, CLA use this information to analysis texture features and make a decision on skin and non-skin regions. Each part of the proposed method is described more in detail next.

3.1 Pixel-based boosted skin detection

There is no explicit evidence which indicates a special color space to have the best performance for SD in all situations and images. The main difficulty in achieving high skin recognition rates, and producing the smallest possible number of false positive pixels, is defining the accurate cluster boundaries through simple (often heuristically chosen) decision rules. Our experiments along with several previous works which have been conducted in this area confirm this fact. Thus, by combining different skin color spaces in an appropriate manner, we aim at obtaining a model which is near to optimum in various situations. The combination of classifiers has been widely studied as a method for improving the efficiency and accuracy achieved by a single classifier [77]. Some combinations, such as those based on the Bayes theory, assume that the classifiers are statistically independent. Voting methods present another type of combination that do not require independency and thus can simply count the number of classifiers that agree with their decision and accordingly decide on the class to which the input pattern belongs.

In our approach, we assume a structure of classifiers in which each classifier is an explicit boundary skin detector in a specific color space. A boosting method called "unbiased voting" [78] is used to combine the classifier results to make a better decision. We use the RGB, YC_bC_r, HSI, Nrgb, and YIQ color spaces in our boosting structure. In fact, in [79,80] these color spaces are reported as good candidates to be used in explicit SD methods.

Our proposed pixel-based boosted SD method assigns a weight to the output of each classifier. This weight indicates the effect of each classifier result on the final decision. The conducted experiments show a direct relation between the *true positive* (TP) and *false positive* (FP) rates in skin detectors. We have found that if the classifier weights are determined such that a balance is achieved between the TP and FP of each classifier, the best results will be obtained by boosting of classifiers. The structure of our pixel-based boosted skin detector is shown in Fig. 9. The weight of each classifier is determined by

$$w_i = \frac{T_i}{\sum_{j=1}^{m} T_j} \tag{3}$$

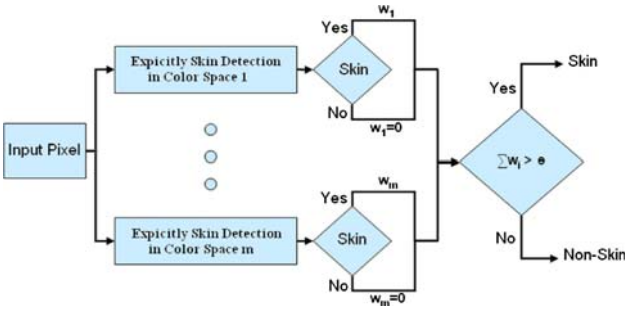

Fig. 9 Structure of pixel-based boosted skin detector

Fig. 10 Two skin candidate regions with different textures

where T_i is the TP rate of each explicit skin classifier on the training data and m is number of explicit classifiers used in the boosted structure. Threshold of pixel-based boosted SD, θ, is determined empirically to detect 94.2% of all skin-associated pixels. Assessment is then made in terms of the percentage of non-skin pixels that are incorrectly accepted. In our experiments, the lowest false acceptance rate was about 25.8%. The threshold can be changed to achieve higher TP versus lower FP. The aim of our proposed method is to detect the early skin regions with high TP and FP and then to reduce FP by employing the texture analyzer.

3.2 Texture extraction

To extract skin texture two facts should be considered. The texture of skin regions is somewhat smooth and no specific texture feature can be recognized from a distant skin region. Note that pixel-based approaches judge whether a pixel belongs to skin based on its color appearance, while our region-based approach combines the color and texture information of skin region candidates to make its decision. As a result, a non-skin region (with a skin-like color) can be discriminated from the obtained set of skin-like regions by considering its different texture properties. Figure 10 shows an example of two skin candidate regions with different textures.

We use a combination of two approaches to extract texture features. In the first approach, we use the Euclidian distance in a given color space to assess neighboring relations. The output of this approach generates a probability map in which

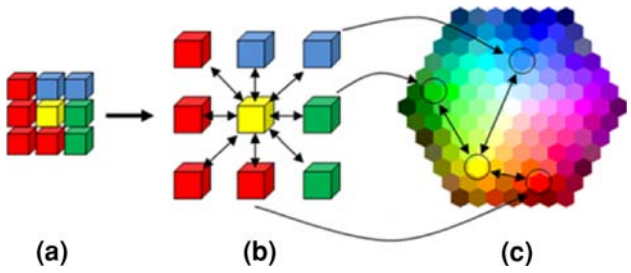

Fig. 11 Calculation of RGB distances. **a** Input color image and its **b** 3×3 adjacency, **c** RGB distances

the corresponding skin pixels have a high probability. In the second approach, we use the difference between the original image and its median filtered image to obtain a coarseness map. We use these two maps to obtain the final probability map that is fed to the texture analyzer. The details of these approaches are given in Sects. 3.2.1 and 3.2.2. In Sect. 3.2.3, the combination strategy of two texture maps to achieve the final probability map is described.

3.2.1 Approach 1

In the first texture extraction method, the texture of a region is extracted based on the color information of its neighboring pixels and their Euclidian distances in the RGB color space. First, for each pixel $X_{ij}(r, g, b)$ located at (i, j) position of image "I" (with red, green, and blue values "r", "g", and "b", respectively) the Euclidian distances between the color of that pixel and all its neighbors in a $W \times W$ block around it is calculated. Note that image the result of pixel-based boosted SD process. Figure 11 shows the calculation of RGB distances for a typical adjacency. The RGB distances are calculated by

$$D_{YX_{ij}} = \sqrt{(r_{X_{ij}} - r_Y)^2 + (g_{X_{ij}} - g_Y)^2 + (b_{X_{ij}} - b_Y)^2} \quad (4)$$

The "coarseness map" is then generated by

$$C_{ij} = \text{Variance}\left(D_{YX_{ij}}\right) Y \in W \times W \text{ around } X_{ij} \quad (5)$$

The "probability map" is then computed using the calculated coarseness value by

$$P_{ij} = 1 - \frac{C_{ij}}{\text{Max}(C_{mn})} \quad m, n \in I \quad (6)$$

Figure 12 shows a typical image and its related probability maps for 3×3 adjacency. Brighter pixels indicate higher skin probability. From this figure, one can observe that skin-like regions which contain a coarse texture have small and irregular probability maps while smooth skin-like regions have higher and more regular probability maps.

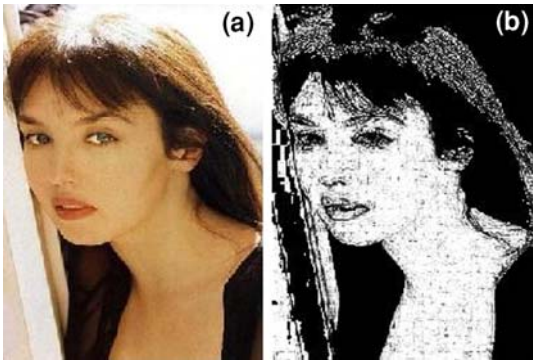

Fig. 12 **a** Original image. **b** Resulted probability maps (for 3 × 3 adjacency) obtained by Approach 1

Fig. 13 **a** Original image. **b** Resulted probability maps obtained by Approach 2

3.2.2 Approach 2

In the second approach, median filtering is used to obtain smoothed textures. The median filter is applied on regions which are classified as skin in the previous pixel-based boosted SD process. This method is similar to the texture extraction algorithm proposed in [22,51]. The grayscale masked image is used to compute the texture data. The intensity of gray image is smoothed with a median filter and the result is subtracted from the masked gray image. Absolute values of these differences are fed to the second median filter. The result must be converted to a probability map to be combined with the previously obtained texture map. The probability map is obtained by

$$P_{ij} = 1 - \frac{M_{ij}}{\mathrm{Max}(M)} \tag{7}$$

where $M_{i,j}$ is the resulting value at location (i, j). Figure 13 shows a typical image and its related probability maps using a 5 × 5 median filter.

3.2.3 Combination of probability maps

In this section, the combination strategy of the two probability maps yielded from previously described approaches is described to achieve the final probability map. To combine these two maps, we use a weighted average of them using

$$\begin{aligned}\mathrm{ProbabiltyMap} = \omega \times \mathrm{ProbabiltyMap1} + (1-\omega) \\ \times \mathrm{ProbabiltyMap2}\end{aligned} \tag{8}$$

in which weight $0 \leq \omega \leq 1$ is obtained by examining different values ranging from [0 to1] (the value which results in a high TP and a low FP is chosen as a proper candidate for ω). Figure 14 shows the result of the combination method for some typical images and their resulted maps. The final probability map is fed to a CLA-based texture analyzer to make a decision based on the overall texture information. The detail of texture analysis stage is explained next.

3.3 CLA-based texture analysis

In this section, CLA is used to make the final decision on skin or non-skin regions using their obtained probability maps. The main idea for using CLA is to use the neighborhood relation of skin-like regions for making better decisions and thus to improve the SD performance. In fact, CLA can propagate the skin probability of neighboring regions to all directions and then by means of this propagation can make a better decision based on the overall available texture information.

To do so, first the 2-D CLA is created with the same dimensions as the input image (using the 3×3 Moore adjacency). Then, a dynamic structural LA is allocated to each cell of CLA, where each LA takes two actions according to skin and non-skin regions. The initial probability associated with CLA is obtained from the final probability map. LA selects each action based on its probability. Then, each learning automaton is rewarded or penalized based on the selected action of the central LA and its neighbors. A rule analyzes the selected action of adjacent automata and makes a decision to reward or penalize the central LA.

CLA performance is very sensitive to some parameters such as α and β in the learning process and the adjacency window size. In most applications, these parameters are considered to be fixed during the iterations. But, fixed values do not perform well in all states. Consequently, we have let these parameters to change dynamically to improve the CLA performance. In fact for SD, in the early iterations the adjacent cells in CLA do not have much information exchange among them and therefore have almost no information. Thus, at first, a large adjacency window is chosen which causes more LA to contribute to the decision making process. Then, after some iteration, the adjacent LAs will gain more accurate

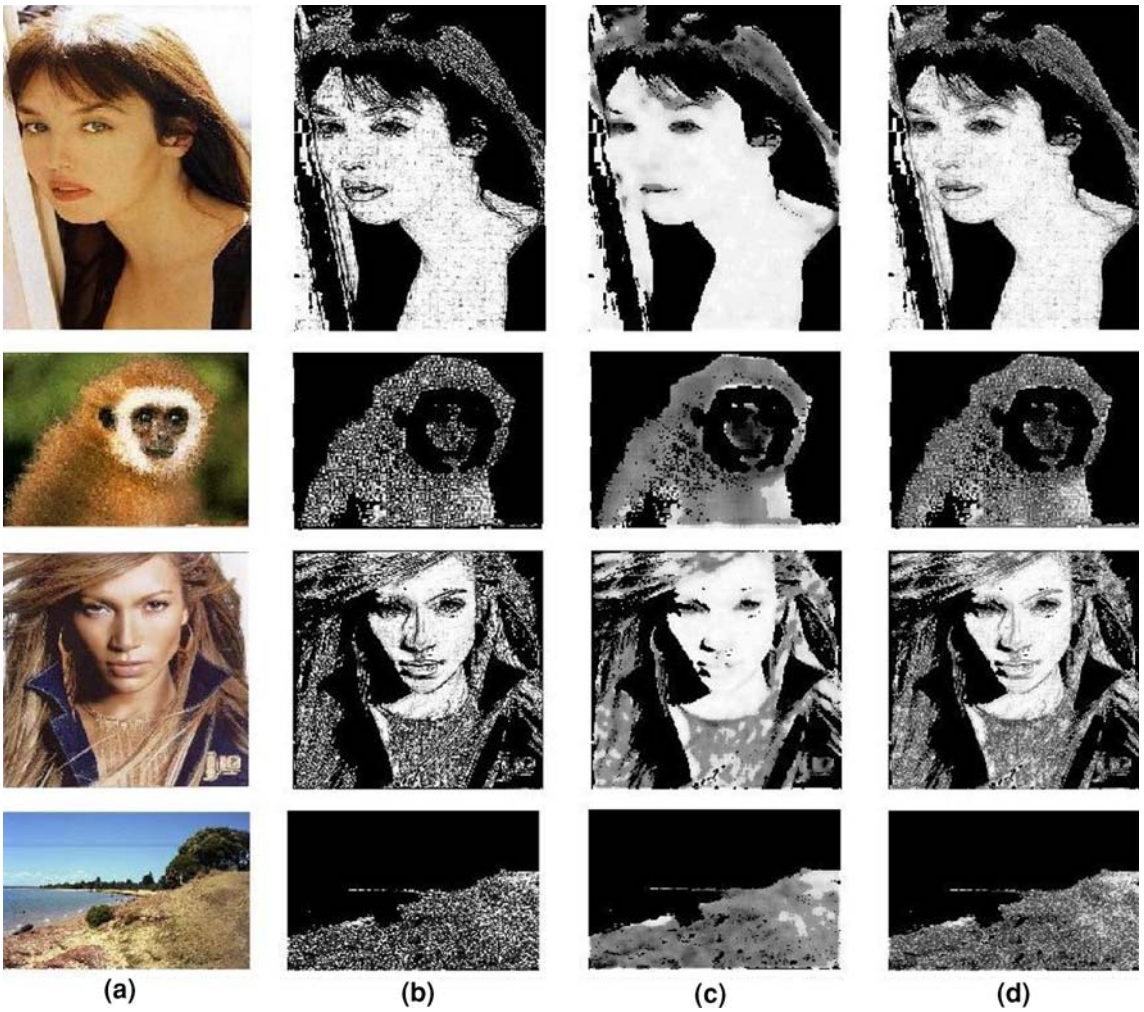

Fig. 14 **a** Original image and the resulted probability maps obtained from: **b** Approach 1, **c** Approach 2, **d** combination process

information (because they have exchanged their knowledge) and hence the window size is reduced.

Moreover, the lack of information can affect the rewarding and penalizing policy and hence reduce the CLA efficiency. Therefore, in the early iterations the rewarding and penalizing must be performed with a low risk. To do so, parameters α and β initially must be chosen with low values. This decreases the influence of incorrect decisions on the final results. After some iteration, the knowledge of LA gets more accurate and we can reward or penalize the decisions with higher confidence and thus increase the values of α and β. These tasks increase the convergence speed of CLA. The relation of changing α and β values with the number of iterations is modeled by

$$\begin{aligned} \alpha_{\mathrm{Cur}} &= \alpha_{\mathrm{Min}} + \left(\frac{\alpha_{\mathrm{Max}} - \alpha_{\mathrm{Min}}}{I_{\mathrm{Max}}}\right) \times I_{\mathrm{Cur}} \\ \beta_{\mathrm{Cur}} &= \beta_{\mathrm{Min}} + \left(\frac{\beta_{\mathrm{Max}} - \beta_{\mathrm{Min}}}{I_{\mathrm{Max}}}\right) \times I_{\mathrm{Cur}} \end{aligned} \tag{9}$$

in which α_1 and β_1 are initial values of α and β, respectively, α_{Max}, α_{Min}, β_{Max}, and β_{Min} are the maximum and minimum acceptable values of α and β, I_{Max} is the number of maximum CLA iteration, and I_{Cur} is the number of current iteration. Also, α_{Cur} and β_{Cur} are the values of α and β in the current iteration. The adjacency window size is changed dynamically between W_{Min} and W_{Max} using

$$W_{\mathrm{Cur}} = W_{\mathrm{Max}} - 2 \times \left[\frac{(K+2) \times (I_{\mathrm{Cur}} - 1)}{I_{\mathrm{Max}}}\right] \tag{10}$$

where W_{Min} and W_{Max} are the maximum and minimum size of the adjacency window, respectively, and K is the number of odd values between W_{Min} and W_{Max}. Note that W_{Min} and W_{Max} must be initialized with odd values to define a legal adjacency around the central pixel. Equation (10) indicates that the adjacency window size changes dynamically from W_{Min} to W_{Max} with decreasing factor 2. For Example, if $W_{\mathrm{Max}} = 7$ and $W_{\mathrm{Min}} = 3$, then K will be 1 and W changes

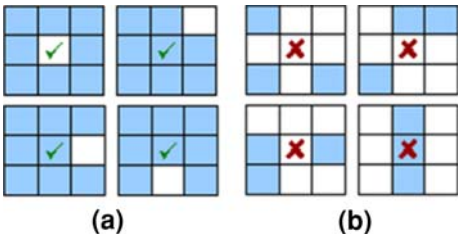

Fig. 15 Examples of **a** reward and **b** penalize states

dynamically after each 25 iterations from 7 to 5 and then to 3.

The rule which controls the rewarding and penalizing actions in an adjacency window is as follows. In each $W \times W$ Moore adjacency, if the numbers of LA which select the action related to skin is greater than 70% of all LA enclosed in the $W \times W$ window, the central LA is rewarded by means of its learning process. In other words, if more than 70% of all LA enclosed in a $W \times W$ window, report a weak texture region (high probability value in probability map) around it, the region around the central LA can be a candidate skin region. Also, if the numbers of LA that select an action related to non-skin are lower than 40% of all LA enclosed in the $W \times W$ window, the central LA is penalized by its learning process. In other words, if less than 40% of all LA enclosed in a $W \times W$ window report a strong non-skin texture region (low probability value in probability map) around it, the region around the central LA can be a non-skin region candidate. Figure 15 shows some states in which the central automata must be rewarded or penalized in a 3×3 window.

An example of this rule is as follows. In each 3×3 Moore adjacency, if the numbers of LA which select an action related to skin is greater than 6, reward the central LA (i.e., if more than 6 automata report a weak texture region around it, the region around the central LA can be a candidate skin region). If the numbers of LA that select an action related to non-skin is lower than 4, penalize the central LA by its learning algorithm (i.e., if less than 4 automata report a strong non-skin texture region around it, the region around the central LA can be a non-skin region candidate).

After several iterations, the existing automata in CLA converge to select the skin or non-skin state with a high probability (i.e., the skin and non-skin probability associated with each LA converges to 0 or 1). Thus, after several iterations, the overall system converges and the process stops. Now, each pixel can be classified as a skin or non-skin by applying a threshold on its probability. Figure 16 shows the convergence of skin probability map obtained by CLA and its thresholding on some typical images. In this figure the change in probability map is shown for some determined CLA iterations.

4 Experimental results

Proposed SD method was carried out on a 2- GHz processor with 1024 MB RAM on Windows XP professional platform. We have used MATLAB 7.1 package and its image processing toolbox 5.0.2. Experiments were conducted using the standard Compaq database (13640 web images, 4675 skin, and 8965 non-skin images) [27].

In [14], we suggested a simpler version of the CLA-Based SD method which was running under some fixed learning parameters and adjacency window sizes. Here, to improve the performance, we have applied a combined texture extractor and the CLA which benefits from dynamic learning parameters. The provided ROC plots show the superiority of the proposed method. In the pixel-based boosted SD process, we used five color spaces RGB, YC_bC_r, HSI, Nrgb, and YIQ. Table 1 shows the used color spaces, their explicit rules for SD, and their weights, W_i, in the boosted structure. Also, Fig. 17 shows the obtained ROC of the proposed pixel-based boosted skin detector for different thresholds θ. The threshold θ was set to 0.65 to detect 94.2% of all skin-associated pixels with false positive rate 25.8%.

Texture extractor in Approach 1 was applied with a 3×3 window size and in Approach 2 the first and second median filters had 5×5 and 7×7 window sizes, respectively. Table 2 shows the configuration parameters of texture analysis process. The learning algorithm of each dynamic structural LA considered the $L_{R-\varepsilon I}$ algorithm with the Moore neighborhood. The results of these configurations are shown in Fig. 18. In this figure, the ROC curves of the proposed method are shown for some different I_{Max} values (25, 50, 75, and 100) when all discussed parameters (ω, α_{Cur}, β_{Cur}, and W_{Cur}) are changing dynamically. As can be seen in this figure, different iteration numbers, I_{Max}, result in different ROC curves. It is because I_{Max} is an important parameter which affects the dynamic CLA parameters (α_{Cur}, β_{Cur}, and W_{Cur}) and changes the final result considerably. For example, $I_{\text{Max}} = 100$ causes the step size, α_{Cur}, and window size, W_{Cur}, to change to 0.0002 and 33, respectively. And, $I_{\text{Max}} = 25$ causes the step size and window size to change to 0.0008 and 8, respectively.

As can be seen in Fig. 18, 75 iterations of CLA result in a superior ROC. In fact, the reason of low performance of other number of iterations is the influence of other dynamically chosen parameters on the behavior of CLA. It shows that $I_{\text{Max}} = 75$ and $\omega = 0.6$ are proper parameters for CLA and texture combination, respectively. This configuration achieves TP rate of about 86.3% and FP rate of about 9.2% on Compaq skin database which shows its efficiency. Note that there are many values which can be selected for parameters α_{Min}, α_{Max}, β_{Min}, and β_{Max} from a continuous range. As we aim at performing a slow learning for confident CLA convergence, these parameters are selected to be small and vary in a small

Fig. 16 Convergence of CLA for skin probability map. **a** Original image. **b** Combined probability map. Probability map after: **c** 25 CLA iterations, **d** 50 CLA iterations, **e** 75 CLA iterations, **f** 100 CLA iterations

Table 1 Used color spaces and their explicit rules and weights in the boosted structure

Color space	Explicit skin detection rule	Weight in boosted structure (W_i)
RGB [29]	$R > 95$ and $G > 40$ and $B > 20$ and $\text{Max}\{R; G; B\} - \text{Min}\{R; G; B\} > 15$ and $\lvert R - G\rvert > 15$ and $R > G$ and $R > B$	0.2143
YC_bC_r [30]	$Cb \in [77, 127]$ and $Cr \in [133, 173]$	0.2122
HSI [81]	$H \in [0, 50]$ and $S \in [0.23, 0.68]$	0.2120
Nrgb [20]	$\frac{r}{g} > 1.185$ and $\frac{r \times b}{(r+g+b)^2} > 0.107$ and $\frac{r \times g}{(r+g+b)^2} > 0.112$	0.1802
YIQ [23]	$I \in [0,50]$	0.1813

range. Higher values of these parameters result in immature and unreliable convergence of CLA. However, very small values of these parameters increase the convergence time of CLA. The adjacency window size W_{Cur} can also take various odd values greater than 1. The dynamic range of W_{Cur} is limited to $[W_{\text{Min}}, W_{\text{Max}}]$. The large W_{Cur} decreases the TP rate, because it does not consider small skin-like regions and segments them as non-skin regions. Thus, the parameters W_{Min} and W_{Max} are configured so that W_{Cur} be 3, 5, and 7.

As mentioned in Sects. 3 and 4, many parameters contribute to producing of the final SD result. In fact, our conducted experiments show that although we have not claimed that the dynamically chosen parameters of the CLA reach the global optimum values in the parameter space (to achieve the highest TP versus the lowest FP), but it leads to a promising suboptimal point in the solution space.

Table 3 lists the performance of the proposed methods and some previous work which were reported on this dataset in terms of TP and FP rates. Unfortunately, we could not

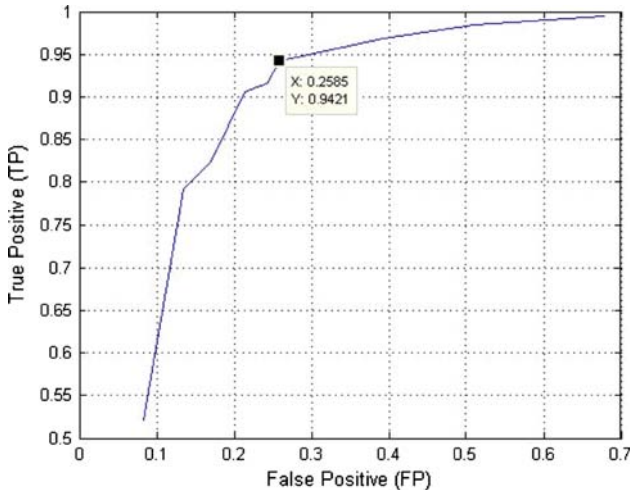

Fig. 17 ROC curve of proposed pixel-based boosted skin detection

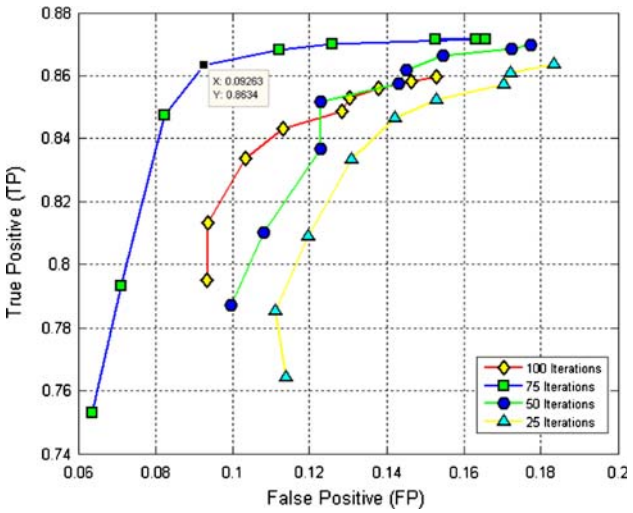

Fig. 18 ROC of proposed method with different number of maximum CLA iterations, I_{Max}, and probability map weights ω

Table 2 Configuration parameters of CLA

Parameter	Value
α_{Min}	0.04
α_{Max}	0.06
β_{Min}	0.02
β_{Max}	0.03
I_{Max}	75
W_{Max}	7
W_{Min}	3

find any region-based skin detectors applied on Compaq skin database for comparison. Thus, we have compared our results with some skin detectors which have reported their results on Compaq. As shown in this table, the usage of texture analysis and dynamic CLA has reduced the FP rate of the pixel-based skin detector but has also decreased the TP rate of it. Figure 19 shows the result of the proposed method on some typical images of Compaq skin database. The elapsed time of the proposed method was about 8 s for 288 × 352 input images which contain skin pixels of about 60%, in average.

5 Conclusions and future work

The main aim of this article was the classification of each query skin-like pixel as a skin or non-skin pixel based not only on its color appearance but also on the existing texture around that pixel. We proposed a skin detection algorithm which extracts some raw information from the color and texture of skin-like regions. This raw information was then fed to a dynamic cellular learning automaton that was able to converge to a stable state based on the related color and texture information. Skin regions were then detected after the convergence of CLA. Our pixel-based method did not consider the texture information of skin-like regions and made a decision purely based on the color appearance of that pixel. The result of skin detection was then fed to a texture analyzer which performed based on CLA and statistical information of skin-like regions. Previous works achieve a high TP rate but with a high FP rate (as shown in Table 3), but our proposed skin detector obtains candidate skin regions with a TP of about 86.3% versus a FP rate of 9.2%. One of the benefits of the proposed method compared with the existing methods

Table 3 Performance of different skin detector methods (on Compaq database)

Detection method	Pixel-based (P) / Region-based (R)	Color space	TP	FP
Bayes [82]	P	RGB	93.4	19.8
Elliptic model [42]	P	Xyz	90	20.9
Thresholding [22]	P	YCbCr	82	18.7
MaxEntropy model [43]	P	RGB	82.9	10
GMM (16 mixtures) [27]	P	RGB	90	15.5
Proposed boosted	P	RGB, YIQ, HSI, Nrgb, YC_bC_r	94.2	25.8
CLA-Based [14]	R	RGB, YIQ, HSV, Nrgb, YC_bC_r	83.4	11.3
Proposed dynamic CLA	R	RGB, YIQ, HSI, Nrgb, YC_bC_r	86.3	9.2

Fig. 19 Obtained skin regions by proposed method: **a** and **d** input images, **b** and **e** pixel-based results, **c** and **f** dynamic CLA-based results

which are based on texture features is that in those methods the decision on skin regions at each block is made by using only the information of that block and its surrounding blocks, but in the proposed method the information of the whole image is counted for to make the final decision. Employment of CLA-based texture analysis and dynamic changes of parameters are the main contributions of this work. Moreover, the boosted pixel-based skin detection can be replaced by any other pixel-based skin detection method, as well.

References

1. Wang, D., Ren, J., Jiang, J., Ipson, S.S.: Skin detection from different color spaces for model-based face detection. Advanced Intelligent Computing Theories and Applications with Aspects of Contemporary Intelligent Computing Techniques, vol. 15, part 14, pp. 487–494. Springer, Berlin (2008)
2. Huang, F.J., Chen, T.: Tracking of multiple faces for human-computer interfaces and virtual environments. In: Proc. IEEE Int. Conf. Multimedia and Expo, New York, pp. 1563–1566 (2000)

3. Ho, W., Watters, P.: Statistical and structural approaches to filtering internet pornography. In: Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 4792–4798 (2004)
4. Sebe, N., Cohen, I., Gozman, F.G., Gevers, T., Huang, T.S.: Learning probabilistic classifiers for human-computer interaction applications. Multimed. Syst. Special Issue Syst. Archit. Multimed. Inf. Retr. **10**(6), 484–498 (2005)
5. Park, S., Aggarwal, J.K.: A hierarchical Bayesian network for event recognition of human actions and interactions. In: ACM Multimedia Systems Journal, Special Issue on Video Surveillance, pp. 164–179 (2004)
6. Yang, B., Hurson, A.R.: Similarity-based clustering strategy for mobile ad hoc multimedia databases. Mobile Inf. Syst. **1**(4), 253–273 (2005)
7. Available online at http://www.idigitalemotion.com/tutorials/guest/skin_tone/skintone.html
8. Shin, M.C., Chang, K., TSAP L.V.: Does color space transformation make any difference on skin detection? In: Proc. IEEE Workshop on Applications of Computer, USA, pp. 275–279 (2002)
9. Albiol, A., Torres, L., Delp, E.J.: Optimum color space for skin detection. Int. Conf. Image Process. (ICIP) **1**, 122–124 (2001)
10. Pung, S.P., Bouzerdoum, A., Chai, D.: Skin segmentation using color pixel classification: analysis and comparison. IEEE Trans. Pattern Anal. Mach. Intell. **27**(1), 148–154 (2005)
11. Vezhnevets, V., Andreeva, A.: A comparative assessment of pixel-based skin detection methods. Technical report, Graphics and Media Laboratory, Moscow State University (2005)
12. Kakumanu, P., Makrogiannis, S., Bourbakis, N.: A survey of skin-color modeling and detection methods. J. Pattern Recognit. **40**, 1106–1122 (2007)
13. Vezhnevets, V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin color detection techniques. GRAPHICON03, pp. 85–92 (2003)
14. Abin, A.A., Fotouhi, M., Kasaei, S.: Skin segmentation based on cellular learning automata. In: Proc. Advances in Mobile Computing and Multimedia, Linz, Austria, pp. 254–259 (2008)
15. Beigy, H., Meybodi, M.R.: Asynchronous cellular learning automata. J. Auomatica **44**(4), 1350–1357 (2008)
16. Meybodi, M.R., Khojasteh, M.R.: Application of cellular learning automata in modeling of commerce networks. In: Proc. 6th Annual International Computer Society of Iran Computer Conference (CSICC). Isfahan, Iran, pp. 284–295 (2001)
17. Dommen, B.J., Croix, D.S.: Graph partitioning using learning automata. IEEE Trans. Comput. **45**, 195–208 (1996)
18. Dommen, B.J., Roberts, T.D.: Continuous learning automata solutions to the capacity assignment problem. IEEE Trans. Comput. **49**, 608–620 (2000)
19. Meybodi, M.R., Beigy, H.: A note on learning automata-based schemes for adaptation of BP parameters. J. Neurocomputing **48**, 957–974 (2002)
20. Gomez, G., Morales, E.: Automatic feature construction and a simple rule induction algorithm for skin detection. In: Proc. of the ICML Workshop on Machine Learning in Computer Vision, pp. 31–38 (2002)
21. Littmann, E., Ritter, H.: Adaptive color segmentation: a comparison of neural and statistical methods. IEEE Trans. Neural Netw. **8**(1), 175–185 (1997)
22. Phung, S.L., Bouzerdoum, A., Chai, D.: Skin segmentation using color pixel classification: analysis and comparison. IEEE Trans. Pattern Anal. Mach. Intell. **27**(1), 148–154 (2005)
23. Dai, Y., Nakano, Y.: Face-texture model based on SGLD and its application in face detection in a color scene. Pattern Recognit. **29**(6), 1007–1017 (1996)
24. Zhanwu, X., Miaoliang, Z.: Color-based skin detection survey and evaluation. In: Proc. 12th International Multi-Media Modeling Conference (MMM '06), pp. 143–152 (2006)
25. Jones, M., Rehg, J.: Statistical color models with application to skin color detection. Proc. Int. J. Comput. Vis. **46**, 81–96 (2002)
26. Yang, J., Lu, A., Waibel, W.: Skin-color modeling and adaptation. ACCV98 Hong Kong. China **1352**, 687–694 (1998)
27. Wang, Y., Yuan, B.: A novel approach for human face detection from color images under complex background. Pattern Recognit. **34**(10), 1983–1992 (2001)
28. Sobottka, K., Pitas, I.: A novel method for automatic face segmentation, facial feature extraction and tracking. Signal Process. Image Commun. **12**, 263–281 (1998)
29. Kovac, J., Peer, P., Solina, F.: Human skin color clustering for face detection. Proc. Int. Conf. Comput. Tool **2**, 144–148 (2003)
30. Chai, D., Ngan, K.N.: Face segmentation using skin-color map in videophone applications. IEEE Trans. Circuits Syst. Video Technol. **9**(4), 518–521 (1999)
31. Brown, D., Craw, I., Lewthwaite, J.: A SOM based approach to skin detection with application in real time systems. In: Proc. British Machine Vision Conference, pp. 491–500 (2001)
32. Sigal, L., Sclaroff, S., Athitsos, V.: Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. IEEE Conf. Comput. Vision Pattern Recognit. **2**, 152–159 (2000)
33. Soriano, M., Huovinen, S., Martinkauppi, B., Laaksonen, M.: Skin detection in video under changing illumination conditions. In: Proc. 15th International Conference on Pattern Recognition, vol. 1, pp. 839–842 (2000)
34. Yoo, T.W., Oh, I.S.: A fast algorithm for tracking human faces based on chromatic histograms. Pattern Recognit. Lett. **20**(10), 967–978 (1999)
35. Chai, D., Bouzerdoum, A.: A Bayesian approach to skin color classification in ycbcr color space. IEEE Region Ten Conf. **2**, 421–424 (2000)
36. Menser, B., Wien, M.: Segmentation and tracking of facial regions in color image sequences. In: Proc. SPIE Visual Communications and Image Processing, pp. 731–740 (2000)
37. Kuchi, P., Gabbur, P., Bhat, S., David, S.: Human face detection and tracking using skin color modeling and connected component operators. IETE J. Res. Special Issue Visual Media Process. **48**(3–4), 289–293 (2002)
38. Terrillon JC, Shirazi MN, Fukamachi H, Akamatsu S (2000) Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In: Proc. International Conference on Face and Gesture Recognition, pp. 54–61
39. Caetano, T.S., Olabarriaga, S.D., Barone, D.A.C.: Performance evaluation of single and multiple-Gaussian models for skin-color modeling. SIBGRAPI02, pp. 275–282 (2002)
40. Yang, M., Ahuja, N.: Gaussian mixture model for human skin color and its application in image and video databases. Proc. SPIE: Conf. Storage Retrieval Image Video Databases (SPIE 99) **3656**, 458–466 (1999)
41. Guillamet, D., Vitria, J.: Skin segmentation using non linear principal component analysis. In: Proc. 2nd Catalan Congress on Artificial Intelligence (CCIA'99). Spain, pp. 224–231 (1999)
42. Lee, J.Y., Yoo, S.I.: An elliptical boundary model for skin color detection. In: Proc. Int. Conf. on Imaging Science, Systems, and Technology, pp. 579–584 (2002)
43. Jedynak, B., Zheng, H., Daoudi, M.: Statistical models for skin detection. IEEE Worksh. Stat. Anal. Comput. Vis. **8**, 92 (2003)
44. Phung, S.L., Chai, D., Bouzerdoum, A.: A universal and robust human skin color model using neural networks. IJCNN01 **4**, 2844–2849 (2001)
45. Chen, C., Chiang, S.P.: Detection of human faces in colour images. IEE Proc. Vis. Image Signal Process **144**(6), 384–388 (1997)

46. Sebe, N., Cohen, T., Huang, T.S., Gevers, T.: Skin detection, a Bayesian network approach. ICPR04 **2**, 903–906 (2004)
47. Juang, C.F., Chiu, S.H., Shiu, S.J.: Fuzzy system learned through fuzzy clustering and support vector machine for human skin color segmentation. IEEE Trans. Syst. Man Cybernetics-Part A: Syst. Humans **37**(6), 1077–1087 (2007)
48. Pujol, F.A., Espi, R., Mora, H., Sanchez, J.L.: A fuzzy approach to skin color detection. MICAI 2008: Advances in Artificial Intelligence, vol. 5317, pp. 532–542. Springer, Berlin (2008)
49. Diplaros, A., Gevers, T., Vlassis, N.: Skin detection using the EM algorithm with spatial constraints. IEEE Int. Conf. Syst. Man Cybern. **4**, 3071–3075 (2004)
50. Kruppa, H., Bauer, M.A., Schiele, B.: Skin patch detection in real-world images. Annual Symposium for Pattern Recognition of the DAGM. Springer LNCS 2449, vol. 109, pp. 109–116 (2002)
51. Forsyth, D.A., Fleck, M.: Automatic detection of human nudes. Int. J. Comput. Vis. **32**(1), 63–77 (1999)
52. Xu, Y., Li, B., Xue, X., Lu, H.: Region-based pornographic image detection. In: IEEE 7th Workshop on Multimedia Signal Processing (MMSP'05). Shanghai, China, pp. 1–4 (2005)
53. Buchsbaum, G.: A spatial processor model for object colour perception. J. Franklin Inst. **310**, 1–26 (1990)
54. Forsyth, D.: A novel approach to color constancy. J. Comput. Vis. **5**(1), 5–36 (1990)
55. Brainard, D.H., Freeman, W.T.: Bayesian color constancy. J. Opt. Soc. Am. **14**, 1393–1411 (1997)
56. Nayak, A., Chaudhuri, S.: Self-induced color correction for skin tracking under varying illumination. ICIP03 **2**, 1009–1012 (2003)
57. Strring, M., Koèka, T., Anderson, H.J., Granum, E.: Tracking regions of human skin through illumination changes. Pattern Recognit. Lett. **24**(11), 1715–1723 (2003)
58. Sigal, L., Sclaroff, S., Atlitsos, V.: Skin color-based video segmentation under time-varying illumination. IEEE Trans. PAMI **26**(7), 862–877 (2004)
59. Barnard, K., Funt, B., Cardei, V.: A comparison of computational color constancy algorithms-Part I: theory and experiments with synthetic data. IEEE Trans. Image Process **11**(9), 972–984 (2002)
60. Barnard, K., Martin, L., Coath, A., Funt, B.: A comparison of computational color constancy algorithms-Part II: Experiments with image data. IEEE Trans. Image Process **11**(9), 985–996 (2002)
61. Bergasa, L.M., Mazo, M., Gardel, A., Sotelo, M.A., Boquete, L.: Unsupervised and adaptive Gaussian skin-color model. Image Vis. Comput. **18**(12), 987–1003 (2000)
62. Cho, K.M., Jang, J.H., Hong, K.S.: Adaptive skin-color filter. Pattern Recognit. **34**(5), 1067–1073 (2001)
63. Frisch, A.S., Verschae, R., Olano, A.: Fuzzy fusion for skin detection, vol. 158, pp. 325–336. Elsevier, Science, Fuzzy Sets and Systems (2007)
64. Xiao, K., Danghui, L., Lansun, S.: Segmentation of skin color regions based on fuzzy cluster. In: Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing Hong Kong, pp. 125–128 (2004)
65. George, D.F.J., George, S.E.: Cellular automata cryptography using reconfigurable computing Source. In: Proc. of the 16th International Conference on Developments in Applied Artificial Intelligence, pp. 104–111 (2003)
66. Mitchell, M.: Computation in cellular automata: a selected review. Technical report, Santa Fe Institute, Santa Fe, NM, USA (1996)
67. Packard, N.H., Wolfram, S.: Two-dimensional cellular automata. J. Stat. Phys. **38**, 901–946 (1985)
68. Kari, J.: Reversibility of 2D cellular automata is undecidable. Physica, pp. 379–385 (1990)
69. Tsetlin, M.L.: On the behavior of finite automata in random media. Automat. Remote Control **22**(10), 1210–1219 (1961)
70. Beigy, H., Meybodi, M.R.: A mathematical framework to study the evolution of cellular learning automata. Adv. Complex Syst. **7**(3–4), 295–319 (2004)
71. Narendra, K.S., Wright, E.A., Mason, L.G.: Application of learning automata to telephone traffic routing and control. IEEE Trans. Sys. Man. Cybern. **7**(11), 785–792 (1977)
72. Oommen, B.J.: A learning automata solution to the stochastic minimum-spanning circle problem. IEEE Trans. Syst. Man. Cybern. **16**, 598–603 (1986)
73. Oommen, B.J., Raghunath, G.: Automata learning and intelligent tertiary searching for stochastic point location. IEEE. Trans. Syst. Man. Cybern. Part B. Cybern. **28**(6), 947–954 (1998)
74. Thathachar, M.A.L., Sastry, P.S.: Learning optimal discriminant functions through a cooperative game of automata. IEEE Trans. Syst. Man. Cybern. **7**(1), 73–85 (1987)
75. Meybodi, M.R., Kharazmi, M.R.: Application of cellular learning automata to image processing. J. Amirkabir **14**(56), 1101–1126 (2004)
76. Beigy, H., Meybodi, M.R.: Open synchronous cellular learning automata. Adv. Complex Syst. **10**(4), 527–556 (2007)
77. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **20**, 226–239 (1998)
78. Haykin, S.: Neural Networks: a Comprehensive Foundation. Prentice Hall PTR Upper Saddle River, NJ (1998)
79. Abadpour, A., Kasaei, S.: Pixel–based skin detection for pornography filtering. Iranian J. Electr. Electron. Eng. **1**(3), 21–41 (2005)
80. Gasparini, F., Corchs, S., Schettini, R.: Pixel-based skin colour classification exploiting explicit skin cluster definition methods. In: Proc. 10th Congress of the International Colour Association, vol. 1, pp. 543–546 (2005)
81. Heieh, I.S., Fan, K.C., Lin, C.: A statistical approach to the detection of human faces in colour nature scene. Pattern Recognit. **35**, 1583–1596 (2002)
82. Brand, J., Mason, J.: A comparative assessment of three approaches to pixel level human skin detection. ICPR01, pp. 1056–1059 (2000)