Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 27 (2006) 1342–1352

Pattern Recognition Letters

www.elsevier.com/locate/patrec

An adaptive real-time skin detector based on Hue thresholding: A comparison on two motion tracking methods

Farhad Dadgostar *, Abdolhossein Sarrafzadeh

Institute of Information and Mathematical Sciences, Massey University, Albany, Private Bag 102 904, North Shore Mail Centre, Auckland, New Zealand

Received 11 July 2005; received in revised form 31 December 2005
Available online 20 March 2006

Communicated by G. Borgefors

Abstract

Various applications like face and hand tracking and image retrieval have made skin detection an important area of research. However, currently available algorithms are based on static features of the skin colour, or require a significant amount of computation. On the other hand, skin detection algorithms are not robust to deal with real-world conditions, like background noise, change of intensity and lighting effects. This situation can be improved by using dynamic features of the skin colour in a sequence of images. This article proposes a skin detection algorithm based on adaptive Hue thresholding and its evaluation using two motion detection technique. The skin classifier is based on the Hue histogram of skin pixels, and adapts itself to the colour of the skin of the persons in the video sequence. This algorithm has demonstrated improvement in comparison to the static skin detection method.

Keywords: Adaptive skin detection; Hue thresholding; Frame subtraction; Optical-flow motion tracking; Human–computer interaction

1. Introduction

Hand, head, and body tracking have become important research topics in human–computer interaction (HCI), during the last decade. Moreover, the applications are not limited to HCI, and have many other usages, like animation creation, virtual reality, disability support, performance measurement, and movement analysis. The approaches that require attachment to the body, from the user's point of view considered more intrusive. Therefore, approaches like vision-based methods are preferable from the user's point of view and can be more general.

In research literature there are many studies on vision-based body tracking, detection and recognition. For many applications factors, such as availability of the hardware and real-time operation are considered the most important characteristics. There are two significant approaches for body-parts tracking: pattern recognition (e.g., using neural networks or statistical analysis) which usually are used for face detection (Viola and Jones, 2004), and segmentation (e.g., skin colour segmentation), are used for hand tracking and gesture recognition (Sigal et al., 2004).

This paper presents a real-time approach for skin segmentation based on motion features of a video sequence. The system is reliably operable in a wide range of office environments. It is robust with typical environment lighting and un-calibrated camera. The proposed system needs an initial training with a small number of samples. It retrains and adapts itself to the user's skin colour, while he/she is using the system.

2. Research background

Skin colour segmentation has shown promising results for hand/face detection and tracking (Bretzner et al.,

* Corresponding author. Tel.: +64 9 4140800x9257; fax: +64 9 4418181.
E-mail address: F.Dadgostar@massey.ac.nz (F. Dadgostar).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.01.007

2002; Imagawa et al., 1998), including RGB, ICrCb, HSV, HSI, HS and IUV colour spaces (Chen et al., 2002; Wong et al., 2001, 2003; Zhu et al., 2004). The main idea in these approaches is using a set images that in which the skin region was manually segmented, to find a region in colour space and using the result as a colour skin probability density function. This procedure is called "training" in the literature. The result of training in most of the colour spaces is a connected region. which means the boundaries of the training data in the colour space, can be specified by a limited number of surfaces, lines or points relatively accurate. These boundaries of the favourite region in the space are called thresholds.

Research shows the best results have been achieved by using RGB, HS, and Hue colour spaces (Bradski, 1998; Sigal et al., 2004). Obviously, a larger number of dimensions in colour space requires a larger amount of training data, and memory space. In addition larger number of dimensions requires more thresholds to specify the favourite region.

In some of these colour spaces, intensity is implicit (RGB) or is one of the dimensions (HSI, IUV, ICrCb), which makes a classifier based on that colour model to be vulnerable against intensity changes in the detection phase. For covering wider range of intensities a bigger region in colour space is required, which means more false detections and less accuracy in the final results. Moreover, a smaller region in colour space however reduces the number of false detections but it also reduce the number of correct detections which means a weaker classifier.

HS (Zhu et al., 2004) and Hue (Bradski, 1998) are other colour spaces have that also shown acceptable results for colour skin segmentation. The histogram of the skin colour is a small-connected region in these colour spaces which is almost significant from the histogram of the background pixels. This feature also makes the formulation of the classifier easier and computationally cheaper. In addition, research shows that probability density function in distribution of Hue factor in different skin colours is a small Gaussian-like connected region (Sigal et al., 2004), and for the colour skins of the different races, the Saturation factor is different (Bradski, 1998). In addition Hue colour space has been used successfully for skin segmentation in several researches (Bradski, 1998; Shin et al., 2002; Zhu et al., 2004).

Bradski (1998) has used this method to find the centre of mass of the skin pixels for face detection of the computer user. Kolsch and Turk (2004) have used a similar approach for detecting a group of features they called "flocks of features", for hand tracking. Ruiz-del-Solar and Verschae (2004) have used this technique together with a fuzzy approach for calculating the membership degree of a pixel to the colour skin set based on its probability density and its neighbours' probability density. Imagawa et al. (1998) have used a mixed approach based on locating the face using non-invariant features and estimating the colour probability density function for segmenting hands.

Fig. 1. In some conditions probability density of the skin colour of the user is lower than some of the unwanted regions like wood colour.

All of these methods except Imagawa et al. (1998), cannot dynamically adapt themselves to the changes of colour skin features like slight histogram shifting, due to change of lighting, environmental noise, or the effect of the colour of objects that are similar to skin colour. For instance, the Hue factor of wood, is similar to skin colour, and using it as indicated in (Bradski, 1998), the filtered image using skin segmentation based on Hue factor, will be like Fig. 2b.

Using the histogram as a probability density function, as mentioned in (Ruiz-del-Solar and Verschae, 2004), and calculating membership probability to skin colour based on neighbourhood pixels, can reduce the number of Erode-Dilate morphological operations that are required for noise removal. However, the assumption that the skin colour of the current user has the highest probability in the probability density function is not always reliable. Fig. 1 presents a filtered image based on probability density of pixels. Lighter pixels have highest probability in the training data and darker pixels have the lowest probability. In this condition, the segment that belongs to the skin colour of the user is almost darker than the segment that belongs to the surface of the table. Therefore, using the work of Ruiz-del-Solar and Verschae (2004), the best case scenario would be segmenting the surface of the table as well as the skin region, which means a weak skin detector.

The idea introduced by Imagawa et al. (1998), because of the retraining and adapting to the colour space that is used in the image, is more robust. However, in comparison to other methods, it requires a significant amount of computation for face detection. In addition, most of the fast face tracking techniques (Viola and Jones, 2004) are not robust to changes like rotation or to situations where a complete frontal view of the face is not presented. This weakness can be enhanced by using rotated images in training time and improving the face detection technique (Wu et al., 2004; Zhang et al., 2002). However, using one of these techniques itself requires a considerable amount of computation in detection.

3. The approach

Based on the discussion and the results of other research work discussed in the previous section, we believe that using Hue factor for skin segmentation could be one of the fastest methods for implementing a skin detector. Because it is a one dimensional colour space, therefore, the skin colour region can be specified by two thresholds.

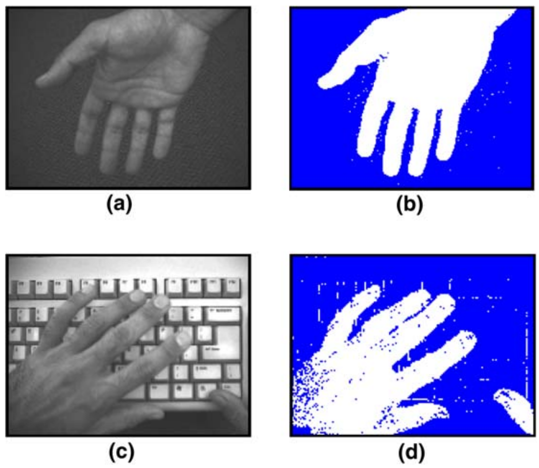

Fig. 2. (a,c) Original image and (b,d) filtered image using Hue thresholding.

It requires few CPU instructions per pixel (Bradski, 1998), that makes it a good candidate for real-time skin detection. In addition, it is reliable for ideal conditions (e.g., special applications, or using a blue background—Fig. 2).

Based on the above description, we developed a static skin detector using Hue thresholding. We called this system *Global Skin Detector* or *GSD*.

However, a GSD can detect the actual skin pixels with reasonable rate, but it also falsely detects some non-skin pixels. In addition, it cannot distinguish those objects that have the similar Hue factor like skin colour (e.g., wood). In some situations even the amount of falsely detected pixels is more that the actual skin pixels, which makes it to be impractical for real-world applications. On the other hand, our observations show that:

1. The peak, including position and height, and width of the training histogram are dependent on the image grabbing hardware, and therefore the best results for a static (separate training and detection) algorithm can only be achieved by using the same hardware.
2. Manually changing the lower and upper bounds of the thresholds can significantly improve the results, and the new thresholds are always inside of the boundaries of Global Skin Detector.

This observation, and the fact that in a video sequence people normally move their head or hands, leaded us to the idea of adapting the Global Skin Detector's thresholds. The information of the skin in the image can be re-evaluated locally using motion features of the image. That means improving the detection of the skin colour through the time as described in the next section.

4. The adaptive skin detection algorithm

The skin detection algorithm originally introduced by the authors (Dadgostar and Sarrafzadeh, 2005; Dadgostar et al., 2005). This algorithm has four main steps as follows:

(1) *Training the global skin detector* using a set of training data and specifying the thresholds of skin colour in Hue colour space. The detailed description of this step introduced in Section 6.1.

(2) *Detection in-motion skin pixels*. The next step is detecting the in-motion pixels of the image, using a motion detection algorithm, and filtering the detected pixels using the Global Skin Detector. The output of this step, are those pixels that with a higher probability belong to the skin regions of the image.

(3) *Recalculating the thresholds*. In the next step, the pixels that were considered as moving pixels belonging to the user's skin are used for retraining the detector. In this article we have used a histogram of Hue factor as the base for calculating low (T_L) and high (T_U) thresholds for filtering the image. From the in-motion skin pixels (Fig. 5c), another histogram is extracted, and the second histogram is merged by the original histogram using the following equation: $H_{n+1} = (1 - A) \times H_n + A \times H_M$.
- H_{n+1} is the new histogram for skin detection (for the next frame).
- H_n is the histogram for skin detection in the current frame.
- H_M is the histogram of the in-motion pixels of the skin colour (Fig. 5c), and uses as a feedback to provide information about the Hue factor of the skin colour of the user.
- And A is the weight for merging two histograms.

Empirical results show that a value between 0.02 and 0.05 brings the best output for the final skin detector. The graph showing the relationship between the merging factor, correct detection and false detection presented in Section 6.3, also approves our empirical results.

(4) *Filtering using adaptive skin detector*. For each frame, thresholds of the Hue factor are recalculated such that they cover 90% of the area of the new histogram. The filter for each frame could be described as follows:

$$f(1) = \begin{Bmatrix} \text{true} & \text{if} & T_L(H_n) \leqslant I \leqslant T_U(H_n) \\ \text{false} & \text{else} & \end{Bmatrix} \quad (1)$$

- I is the Hue factor for each pixel.
- H_n is the Hue histogram for the skin colour.
- T_L is the calculated lower threshold for histogram H_n.
- T_U is the calculated upper threshold for the histogram H_n.

4.1. Summary of the algorithm

Fig. 3 presents the block diagram of the adaptive skin detector.

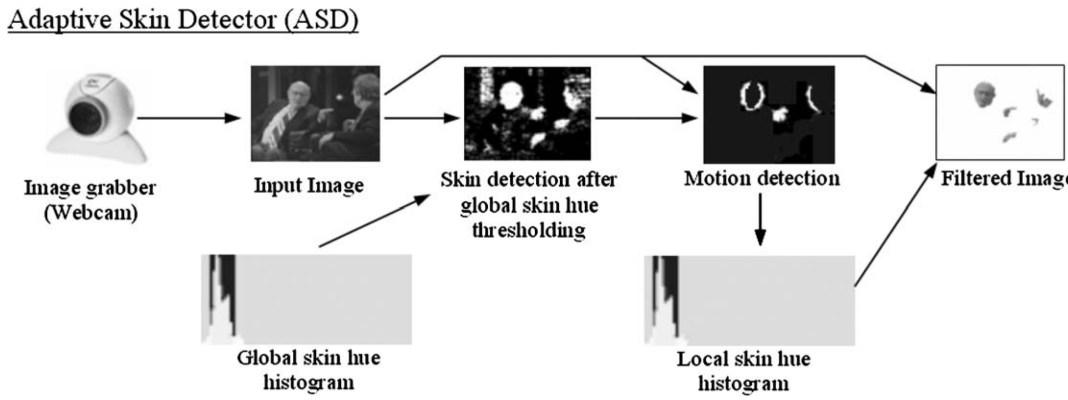

Fig. 3. Overview of the adaptive skin detector.

5. Motion tracking

Motion tracking and detection, attracted considerable research interest in computer vision community, and therefore there are many different approaches for this problem. Depending on the conditions of the application, each of them specified different constraints and robustness, and some of them have shown better results in specific applications, but the suitable method of motion tracking for this algorithm is still an open question. Some of them are based on the old idea of frame subtraction, which probably is the simplest approach for motion detection (Bradski and Davis, 2002; Manzanera and Richefeu, 2004; Richefeu and Manzanera, 2004). Alternately some others using more complex features for tracking the motions (Jayaram et al., 2004; Viola et al., 2003; Yang et al., 2002).

In this article, we have used two of the motion tracking methods, for measuring their effect on the adaptive skin detector algorithm. The first one is the well-known frame subtraction method, and the second one is the optical-flow motion tracking, proposed by Lucas and Kanade (1981).

5.1. Underlying assumptions

We should note that in this study, our primary assumption has been that the background is not moving, the camera is in a fixed position, and the person's body is the only moving object in front of the camera (e.g., Fig. 4).

Fig. 4. Testing environment: (a) camera view and (b) configuration settings.

5.2. Frame subtraction motion tracking

Frame subtraction, is one of the basic methods for motion detection in video sequences. This method is based on comparing corresponding pixels of two frames, and considering those pixels which their difference is more than a certain threshold, as changes pixels. In ideal conditions, changed pixels can potentially belong to a moving object. Although this technique is not reliable for recognizing the moving object itself, but it requires small memory space (one frame) and a small number of operations per pixel (integer subtraction), which makes it suitable for real-time applications.

For comparing two pixels, we tested different components of the pixel for comparison, including RGB, CrCgCb, Hue and Intensity. Our observations show that the most reliable results gained from RGB and Intensity comparison, and results from those components that did not carry intensity information were not reliable for motion detection. In the final implementation of the frame subtractor, we used Intensity component of the pixels for comparison.

In a real application, this technique has two main problems. The first problem is that the noise in CCD cameras can cause some sparse falsely-detected pixels. This effect can easily be eliminated, using simple morphological operations. Another problem is that a moving object in a 2D image fills the space that in the previous frame was the background, and the background fills the pixels that previously were the object. Thus, we have two sets of pixels. One set belongs to the object, and the other set belongs to the background. We solved this problem by ignoring those pixels that cannot pass through the primary filter for skin colour detection. Considering the fact that the body of the subject in the video sequence is the only moving object and the camera is in a fixed position. Therefore, the probability of detecting some parts of the skin will be higher than non-skin (Fig. 5c).

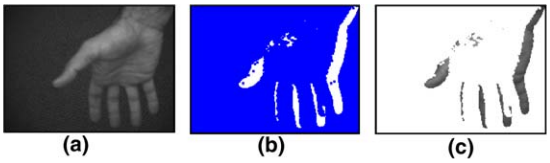

Fig. 5. A moving hand: (a) original image, (b) in-motion pixels of the frame, filtered using Hue threshold and (c) mapping the result to the original image.

5.3. Optical flow motion tracking

The optical-flow motion tracking technique is a more accurate method of object tracking in video sequences. The idea is based on the fact that lighting condition in two sequential frames is relatively constant, and therefore the intensity of each point of an object would be the same. If the position of a specific intensity changes in the next frame, then we can assume that the point has moved to the new place, in the time of in between two frames. Tracking of the segments in optical-flow algorithm is based on the fact that the intensity of a moving segment changes slightly in two sequential frames. Therefore, the next position of a track point will be around its previous position.

Considering the possible movements that the object may have (e.g., moving left, right, up and down, rotation, moving toward or away of the camera), there are several implementations for this technique. In this research we have used an implementation proposed by Lucas and Kanade (1981) which is called Lucas–Kanade optical-flow tracking, and is available in OpenCV (Intel, 2005), as measurement for motion tracking. The implementation is based on choosing a number of tracking points, a small block size to track and a neighbourhood distance to search. These parameters, may affect the processing speed of the algorithm, to make it suitable for real-time applications, we chose 100 track points and block size 5. The proposed optical-flow algorithm has two main parts: initialization track points and tracking.

5.3.1. Initialization track points

Normally, for initializing the optical-flow algorithm and setting the track points, a set of small segments that have intensity different than neighbourhood segments are chosen. Based on this approach most of the track points will belong to the edges in the image. Our observations show that using this approach for tracking skin segments produces poor results. Because of smooth intensity changes of the skin colour of the face, most of track points that belong to skin are located around eyes, nose and mouth, and the number of track points belonging to the skin segments of the face, are very low (Fig. 6a).

To improve the results, we initialized the track points on the skin segments (based on global skin detector) with equal distances (Fig. 6b). This method has two advantages for this application: First of all, we will have a reasonable number of track points on the region of interest, which is the skin colour here. Secondly, by tracking motion, we can find bigger parts of skin segments in comparison to the traditional method.

5.3.2. Tracking

In the initialization section, we put the track points on those regions that are potentially skin. Therefore, recognizing motion in these points reveals user's skin segments with a higher probability. In Fig. 6c, darker points have been recognized as moving skin segments.[1] The small lines connected to these points, represents direction of the movement. Fig. 7, presents the in-motion skin detected by the optical-flow tracking method.

6. The experiment

6.1. Specifying the thresholds of Global Skin Detector

For calculating initial Hue thresholds for skin colour, a set of training data of 20 coloured images (approximately 3,200,000 pixels) of hand in which the skin region had been manually segmented, used. Half of these images were recorded using a Dragonfly digital camera, and the other half collected from the Internet. Using this data, a histogram for Hue factor was calculated and the lower and upper bounds of the threshold specified such that 90% of the pixels inside the histogram were covered. The values of 3 and 43 calculated for lower and upper Hue thresholds, respectively. The range of Hue factor was 0–255 calculated by RGB values of the input pixels.

6.2. Ground-truth data

For evaluating the algorithm, we used three video sequences. Two of them are of the human-human interactions and the third one is human–computer interaction. The length of all of the sequences was 625 frames. The camera was in a fixed position in all of the sequences, there was no movement in the background, and the lighting condition was constant. Table 1, presents the detailed information of the image datasets. The ground-truth data was manually prepared for frames 1–625 by equal distances of 25 frames. Each ground-truth frame was an image containing the skin regions (Fig. 8b).

The first dataset was a video clip of a news interview, the most of the hand and face movements was doing by the person on the left side of the image (Fig. 8a). The second dataset was a chat session between two persons (Fig. 8c). The man on the left side of the image played with a pen using both hands, and the woman on the right side of the image played with a billiard-sized ball. Both subjects have hand movements and slight head movements, while talking. The third dataset was a side view of a computer user,

[1] The video of this implementation is available at: http://www.massey.ac.nz/~fdadgost/xview.php?page=videos.

Fig. 6. (a) Track points based on traditional approach (good features to track), (b) initializing track points, based on primary skin colour filter and equal distances and (c) tracking skin colour segments.

Fig. 7. Nodding the head: (a) original image, (b) in-motion pixels of the frame, filtered using Global Skin Detector and (c) mapping the result to the original image.

while interacting with computer. He had slight head movements to look to the keyboard, and a few hand movements, during the interaction.

For evaluating the algorithm we run the adaptive skin detector on all the three datasets. The adaptive skin detector was running continuously to the end of the experiment, and the evaluation procedure was doing based on the output of the adaptive skin detector comparing to the ground-truth data.

6.3. Measured parameters

We have measured four parameters for each output based on its ground-truth data and for each dataset separately:

(1) *Correct detection ratio* (R_c) is the number of correctly detected pixels to the actual number of skin pixels in the image. The ideal system has a value equal to 1, for this factor.

(2) *False detection ratio* (R_f) is the number of falsely detected pixels to the actual number of non-skin pixels. The ideal system has a value equal to 0 for this factor.

(3) *Performance* (P) of the algorithm defines as follows:

$$P = \frac{R_c - R_f}{1 + R_f} \tag{2}$$

As indicated by this formula, the *performance* is equal to 1, when the $R_c = 1$, and $R_f = 0$, a bigger value for R_f makes the *performance* a smaller number. The ideal system has a performance equal to 1, which means there is no false detection, and 100% correct detection.

(4) And the *unreliability rate* (UR) that is the number of falsely detected pixels to the number of actual skin pixels. This factor presents how much the output is inaccurate (or accurate). For instance a value of 2 means the number of falsely detected pixels is 2 times more than the total number of actual skin pixels. Obviously, a lower number for this factor presents a better output. The ideal value for this factor is zero, which means there is no falsely detected pixel, and 100% correct detection. The reason of introducing this parameter is that the *false detection ratio* cannot represent how much output is inaccurate. For instance it is possible to have a $R_f = 0.1$, which is a small number, but the number of falsely detected pixels to be more than the number of correctly detected

Table 1
The characteristics of the image datasets used for the experiments

	Dataset 1	Dataset 2	Dataset 3
Video recording device	Unknown	Unknown	Dragonfly digital camera[a]
Resolution	384×288	352×288	320×240
Source	Anvil—Gesture analysis toolkit[b]	CMU Test Images—Paris dataset[c]	Prepared by the author
Frame rate	25 fps	25 fps	15 fps
Output image	RGB	RGB	RGB
Number of people in the scene	2	2	1

[a] http://www.ptgrey.com/products/dragonfly/
[b] http://www.dfki.de/~kipp/anvil/
[c] http://www-2.cs.cmu.edu/%7Ecil/v-images.html

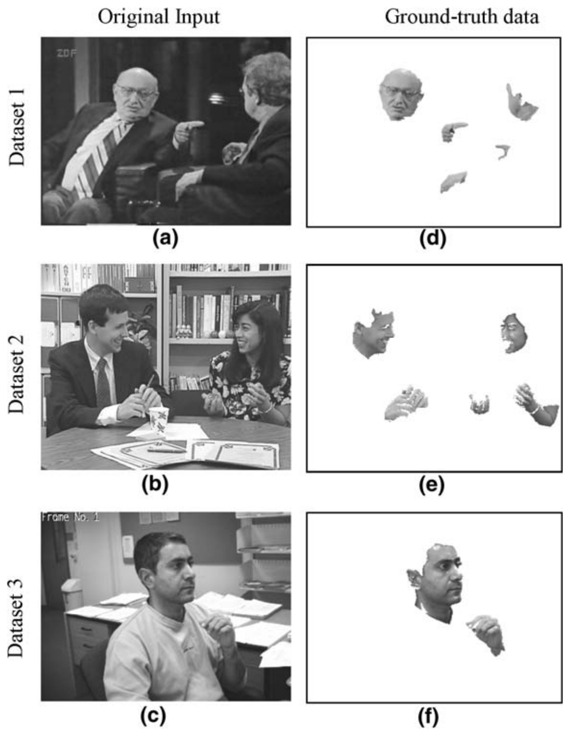

Fig. 8. Samples of the input images and their ground-truth data.

pixels (Fig. 10f), and therefore R_f alone, may give a false impression as accuracy or usability of the output.

6.4. Choosing the merging factor

For indicating the best value for the merging factor of the adaptive algorithm, our empirical results showed that a value between 0.01 and 0.05 produce the best outputs. The experiment (Fig. 9) also confirmed this observation. Increasing the merging factor parameters causes the drop of the both correct and false detections. Obviously, we are interested in keeping the correct detection high, and making the false detection lower. As indicated in Fig. 9, to have a reasonable correct detection ratio, the merging factor should be between 0.01 and 0.06.

Choosing a merging factor of 0, in fact causes the algorithm to be non-adaptive, and to act same as a global skin detector, that however causes the highest number of correct detections but also produce a high number of false detections. On the other hand, choosing a very small number for the merging factor makes the convergence speed of the algorithm, very low. That means reaching to a reasonable output, and adapting to the new conditions of the input (e.g., turning on a light), also will take more time. Also choosing a very small value, because of the round-up problem may have the same effect of a zero value. In the present research we have used a merging factor equal to 0.05 for the experiments, which shows reasonable adaptive speed and stable output.

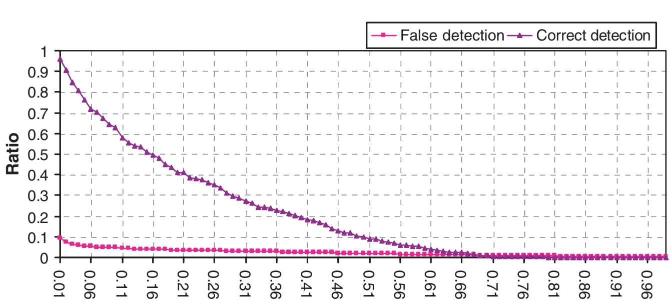

Fig. 9. The accuracy of the algorithm based on merging factor.

7. Results

Filtering the input image using the described method significantly improved the performance of the skin detector, by decreasing the ratio of false detections and keeping the ratio of positive detections, high. In the initial frames, the performance of the adaptive and non-adaptive filter were the same. In Fig. 10g, the surface of the table caused false detection, as the number of its pixels is approximately bigger than positive recognition. The rigid blobs of falsely detected pixels cannot be removed using morphological operations. The last frame's output shows significant improvement in false detection ratio (Fig. 10h).

Fig. 11 presents the behaviour of the algorithm in *correct detection ratio*. An average ratio of 0.8 gained after frame 351, and approximately remained constant. The false detection ratio had indicated in Fig. 11, after frame 351, a false detection ratio approximately equal to 0.05 has indicated and remained constant to the end of the video sequence. The performance of the algorithm, after frame 251, is relatively stable between 67% and 72% that together

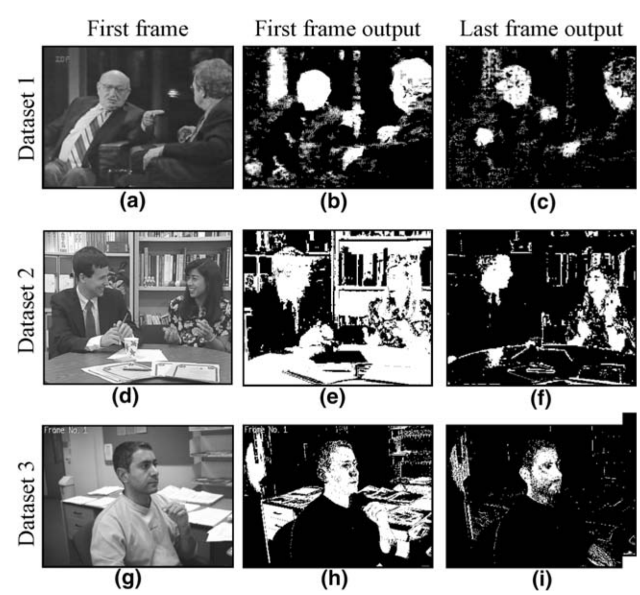

Fig. 10. The changes in output over the time.

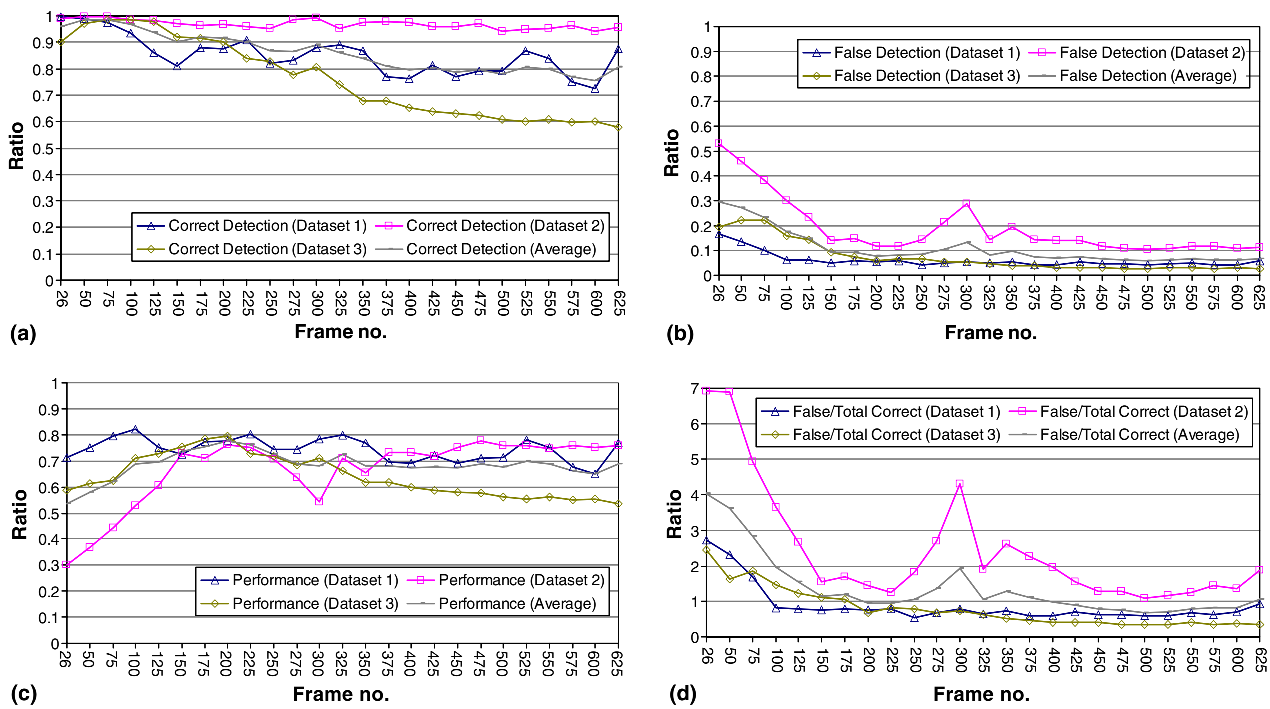

Fig. 11. The behaviour of the adaptive skin detector using frame subtraction motion detection.

with the low ratio of the false detection, represents the stability of the algorithm over time.

The unreliability rate has significantly changed over the time. For instance, the ratio for dataset 2, that initially was 7, dropped to 1 in frame 476, and slowly climbed to 1.8 in frame 601. As represented in Fig. 11d, the output for the dataset 2 initially is very unreliable, but in the last frame output, the detection quality is much better. However, still there are some small falsely detected regions. The reliability rate for dataset 1 and 3 is much better, and after frame 176, was relatively constant. The output for all the three datasets presented in Fig. 11a, and shows improvement.

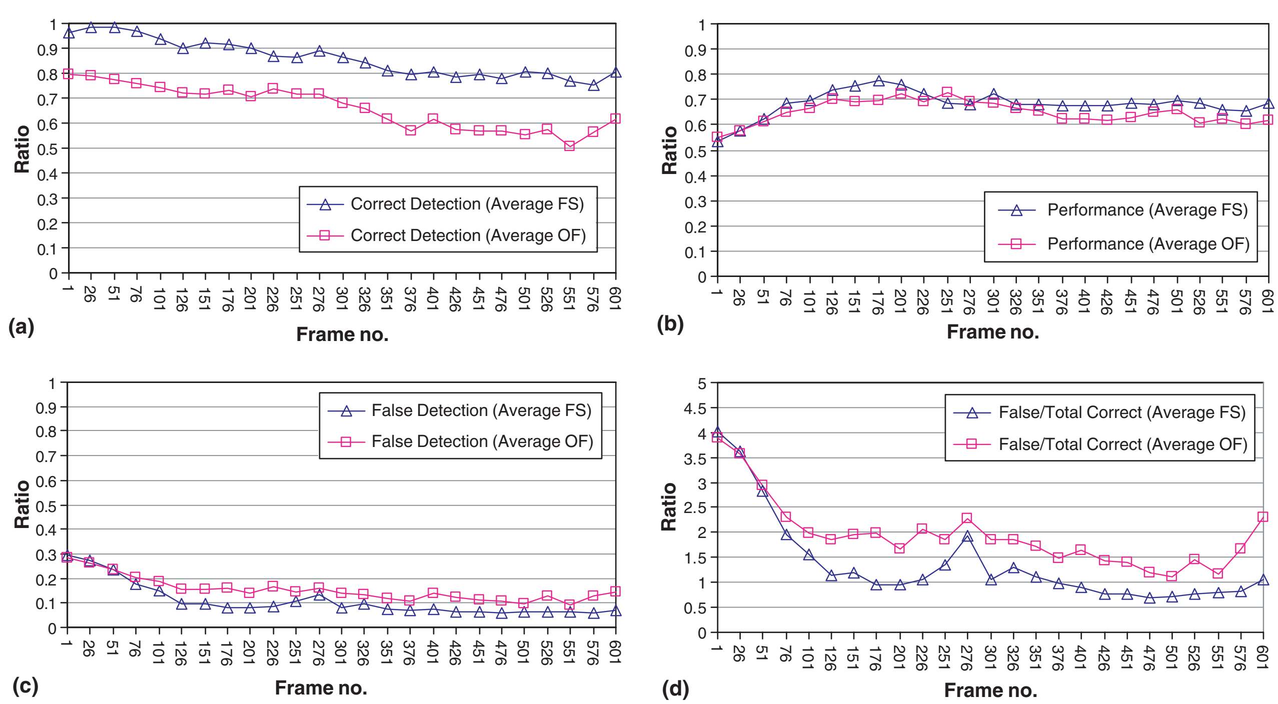

Fig. 12. Comparison of measured parameters of the algorithm, using different motion tracking methods (FS: frame subtraction, OF: optical flow).

7.1. Sparseness of the correctly and falsely detected pixels

For measuring the sparseness of the detected pixels, we used four different morphological operators on the output images of the datasets. Obviously, in case of sparse false detections, the falsely detected pixels can be eliminated using Erode morphological operator. As indicated in Fig. 13, there is no specific pattern on sparseness of the correctly detected pixels.

Fig. 13, shows a shift using different combinations of Erode and Dilate morphological operators. However, using Erode operators in all cases, has dramatically reduced the false detection ratio, which means the falsely detected pixels are more sparse than correctly detected pixels.

7.2. The comparison of the behaviour of the algorithm using different motion tracking techniques

We run another experiment on the algorithm, to verity the effect of different motion tracking methods. The alternate method that has been used here was optical-flow motion tracking, described in Section 5.3.

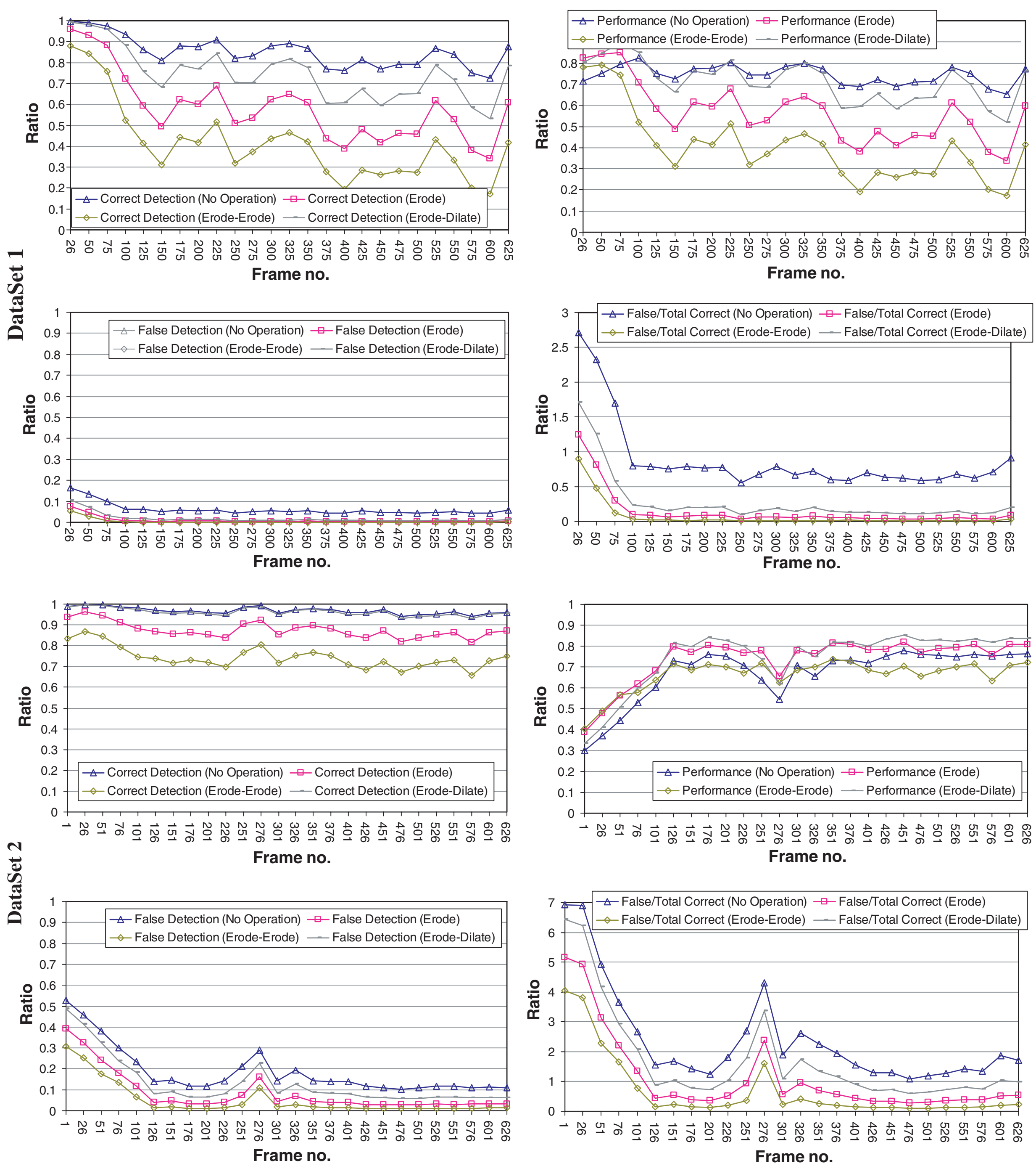

Fig. 13. The results on using morphological operators on all the datasets.

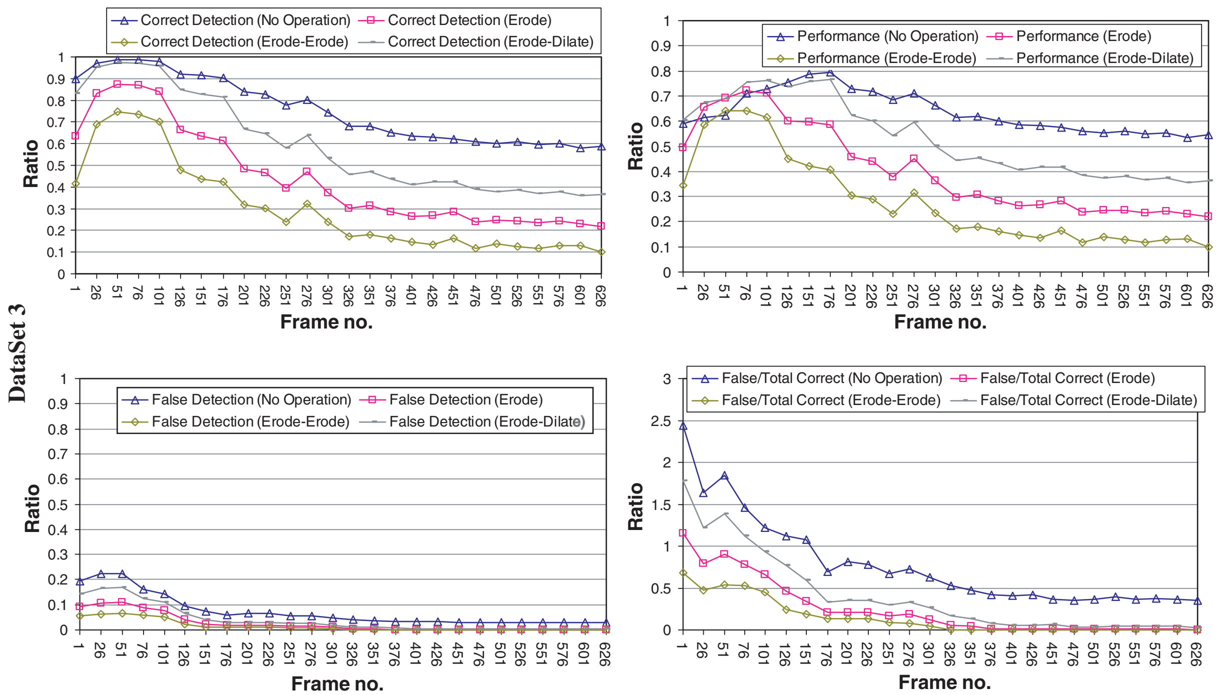

Fig. 13 (*continued*)

The results are presented in Fig. 12. The average correct detection, using optical-flow shows about 20% decrease against frame-subtraction. The average false detection of frame-subtraction is slightly better than optical-flow. The performance of the frame subtraction is slightly better than optical-flow, and the unreliability rate (UR) significantly is better for frame subtraction. The comparison shows, in overall the optical flow motion tracking, as a measurement of motion for the is not as good as frame-subtraction, for this particular algorithm and application.

8. Conclusion

The adaptive skin detector, together with colour skin tracking, retrains itself based on new data gathered from moving pixels. It also recalculates the lower and upper thresholds for filtering the next frame. The proposed algorithm can be used as a component in vision-based applications in human–computer interaction. While the user is interacting with the computer (using mouse, keyboard, and sometimes slight changes in head position), the system captures the movements and retrains itself based on the user's skin colour.

We have used this technique for face tracking the head of the computer user, using the Mean-Shift algorithm (Fig. 14). The application is robust against environmental noise like change of lighting condition, noise caused by fluorescent lights, changes in intensity level caused by automatic gain control in digital cameras, and background noise. The performance of the system is less than 20 milliseconds per frame, on a P4 2.2 GHz PC, for a 24 bit RGB image size 640×480 with non-optimized C++ code.

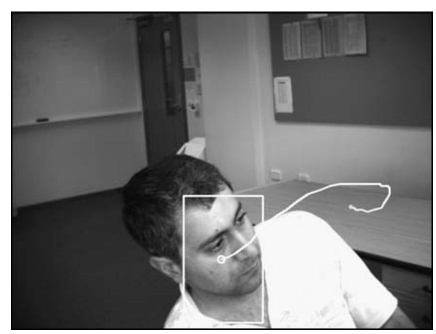

Fig. 14. Face tracking using the proposed algorithm for skin detection, and the Mean-Shift algorithm for blob tracking.

The concept introduced in this article can be applied together with other motion detection algorithms. Using more accurate motion detection algorithms not only can increase the quality of the feedback but also can be used as an extra feature for tracking face and hands.

References

Bradski, G.R., 1998. Computer vision face tracking for use in a perceptual user interface. Intel Technol. J. 2 (2), 1–15.

Bradski, G.R., Davis, J.W., 2002. Motion segmentation and pose recognition with motion history gradients. Machine Vision Appl. 13 (3), 174–184.

Bretzner, L., Laptev, I., Lindeberg, T., 2002. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering, Paper presented at the 5th IEEE Internat. Conf. on Automatic Face and Gesture Recognition.

Chen, L., Zhou, J., Liu, Z., Chen, W., Xiong, G., 2002. A skin detector based on neural network, Paper presented at the IEEE Internat. Conf. on Communications, Circuits and Systems.

Dadgostar, F., Sarrafzadeh, A., 2005. A fast skin detection algorithm for video sequences, Paper presented at the Internat. Conf. on Image Analysis and Recognition (ICIAR), Toronto, Canada.

Dadgostar, F., Sarrafzadeh, A., Johnson, M.J., 2005. An adaptive skin detector for video sequences based on optical flow motion features, Paper presented at the Internat. Conf. on Signal and Image Processing (SIP), Hawaii, USA.

Imagawa, K., Lu, S., Igi, S., 1998. Color-based hands tracking system for sign language recognition. In: Proc. 3rd Internat. Conf. on Face and Gesture Recognition. IEEE Computer Society, Silver Spring, MD, p. 462.

Intel, 2005. Open Computer Vision library (http://www.intel.com/research/mrl/research/opencv/) [Open source library]. Retrieved, from the World Wide Web: http://www.intel.com/research/mrl/research/opencv/.

Jayaram, S., Schmugge, S., Shin, M.C., Tsap, L.V., 2004. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection, Paper presented at the Comput. Vision and Pattern Recognition, 2004 (CVPR 2004), Proc. 2004 IEEE Comput. Soc. Conf. on.

Kolsch, M., Turk, M., 2004. Fast 2D hand tracking with flocks of features and multi-cue integration, Paper presented at the Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW'04), Washington, DC, USA.

Lucas, B., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision, Paper presented at the Proc. 7th Internat. Joint Conf. on Artificial Intelligence (IJCAI).

Manzanera, A., Richefeu, J., 2004. A robust and computationally efficient motion detection algorithm based on Sigma–Delta background estimation, Paper presented at the Indian Conf. on Computer Vision, Graphics and Image Processing, ICVGIP'04, Kolkata, India.

Richefeu, J., Manzanera, A., 2004. A new hybrid differential filter for motion detection, Paper presented at the Internat. Conf. on Computer Vision and Graphics, ICCVG'04, Warsaw, Poland.

Ruiz-del-Solar, J., Verschae, R., 2004. Skin detection using neighbourhood information, Paper presented at the Proc. 6th IEEE Internat. Conf. on Automatic Face and Gesture Recognition.

Shin, M.C., Chang, K.I., Tsap, L.V., 2002. Does colorspace transformation make any difference on skin detection? Paper presented at the Proc. 6th IEEE Workshop on Applications of Computer Vision (WACV).

Sigal, L., Sclaroff, S., Athitsos, V., 2004. Skin color-based video segmentation under time-varying illumination. IEEE Trans. Pattern Anal. Machine Intell. 26 (7), 863–877.

Viola, P., Jones, M.J., 2004. Robust real-time face detection. Internat. J. Comput. Vision 57 (2), 137–154.

Viola, P., Jones, M.J., Snow, D., 2003. Detecting pedestrians using patterns of motion and appearance, Paper presented at the Ninth IEEE Internat. Conf. on Computer Vision, Nice, France.

Wong, K.W., Lam, K.M., Siu, W.C., 2001. An efficient algorithm for human face detection and facial feature extraction under different conditions. Pattern Recogn., USA 34, 1993–2004.

Wong, K.W., Lam, K.M., Siu, W.C., 2003. A robust scheme for live detection of human face in color images. Signal Process.: Image Commun. 18 (2), 103–114.

Wu, B., Ai, H., Huang, C., Lao, S., 2004. Fast rotation invariant multi-view face detection based on real Adaboost, Paper presented at the Automatic Face and Gesture Recognition, 2004, Proc. 6th IEEE Internat. Conf. on.

Yang, M.H., Ahuja, N., Tabb, M., 2002. Extraction of 2D motion trajectories and its application to hand gesture recognition. IEEE Trans. Pattern Anal. Machine Intell. 24 (8), 1061–1074.

Zhang, Z., Li, M., Li, S.Z., Zhang, H., 2002. Multi-view face detection with FloatBoost, Paper presented at the Applications of Computer Vision, 2002 (WACV 2002), Proc. 6th IEEE Workshop on.

Zhu, Q., Cheng, K.-T., Wu, C.-T., Wu, Y.-L., 2004. Adaptive learning of an accurate skin-color model, Paper presented at the Proc. 6th IEEE Internat. Conf. on Automatic Face and Gesture Recognition.