Expert Systems with Applications 39 (2012) 7158–7164

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A comparative study on face detection and tracking algorithms

Rachid Belaroussi [a,*], Maurice Milgram [b]

[a] IFSTTAR, IM, LIVIC, F-78000 Versailles, France
[b] Université Pierre et Marie Curie, ISIR – UMR 7222, 75005 Paris, France

ARTICLE INFO

Keywords:
Face detection
OpenCV
Skin color model
Coupled Camshift

ABSTRACT

Face localization is the first stage in many vision based applications and in human–computer interaction. The problem is to define the face location of a person in a color image. The four boosted classifiers embbeded in OpenCV, based on Haar-like features, are compared in terms of speed and efficiency. Skin color distribution is estimated using a non parametric approach. To avoid drifting in color estimate, this model is not updated during the sequence but renewed whenever the face is detected again, that gives the ability to our system to cope with different lighting conditions in a more robust way. Skin color model is then used to localize the face represented by an ellipse: connected component segmentation and a statistical approach, namely the *coupled Camshift* of Bradsky, are compared in terms of efficiency and speed. The pursuit algorithms are tested on various video sequences, corresponding to various scenarios in terms of illumination, face pose, face size and background complexity (distractor effects).

1. Introduction

Face detection in still images is a hard issue often adressed as a classification problem with two classes: the difficulty is the complexity of defining the non-face class. Since Sung and Poggio (1998), boostrapping approaches have been developed to tackle this problem, most of them were quite successfull in the case of upright faces. A growing research field is concentrating in developing appearance-based model for multi-view and rotation invariant face detection (Wu, Ai, Huang, & Lao, 2004). Multi-view face detection (Schneiderman & Kanade, 2000) aims at detecting faces with out-of-plane rotation (pan and tilt rotation, around the *x*-axis and *y*-axis). Rotation invariant face detection (Rowley, Baluja, & Kanade, 1998) is dedicated to in-plane rotation (head roll, *z*-axis rotation). Both type of pose variation are addressed in Jones and Viola (2003), Li and Zhang (2004), Wu et al. (2004) and Kim, Kee, and Kim (2005). Pose can be estimated by a classifier (Rowley et al., 1998), then the sub-image is derotated and a conventional face detector classifies the candidate. Pose estimation result (Jones & Viola, 2003) can also be cascaded with a N pose specific face detectors. These approaches are powerful in case of still images but still too slow for real-time purpose.

In the case of images sequence, using simple cue such as skin color results in a fast processing and finer face location and pose estimation. Under constraint illumination conditions skin color is robust to variation in scale, orientation and partial occlusion (Phung, Bouzerdoum, & Chai, 2005; Schwerdt & Crowley, 2000; Vezhnevets, Sazonov, & Andreeva, 2003). Lighting conditions can dramatically change during a face tracking process. Difficulties that shall overcome a skin color based tracker are changes in illuminants color (for example when the tracker is used as a desktop application in dark conditions, the light is blue!), non-uniformity of the illumination and skin tones variation across ethnicities. Non-uniformity of the scene illumination can result in shadows on the face - especially when several light sources are present - and in skin color variation when the face is moving across the scene (Soriano, Martinkauppi, Huovinen, & Laaksonen, 2003; Yang & Waibel, 1996). To overcome the luminance effects, several authors implement a face tracking using only the skin color chrominance and dropping the luminance information, in a colorspace were skin color distribution varies smoothly with luminance, and skin color model is updated, from time to time or at each new image across the sequence. Skin color model can be estimated by a parametric (McKenna, Gong, & Raja, 1998) or non-parametric (Swain, 1991) model. Several colorspace are used for skin detection, for example in Bradski (1998) the only Hue channel of HSV is used to estimate skin color distribution, whereas in McKenna et al. (1998), Yoo and Oh (1999) and Zhu, Yang, and Waibel (2000) the HS chrominance plane of HSV is used. Other popular chrominance space include the CbCr plane of YCbCr (Belaroussi, Prevost, & Milgram, 2005; Hu, Worrall, Sadka, & Kondoz, 2004; Seguier, 2004). The ab plane of perceptually uniform color system such as Lab (Li, Goshtasby, & Garcia, 2000; Schumeyer & Barner, 1998) or the uv plan of Luv (Yang & Ahuja, 1999). They all make the assumption that in these chrominance space, the skin distribution is well modeled whatever the skin tones type (african, brown, asian, caucasian).

[*] Corresponding author.
E-mail addresses: rachid.belaroussi@gmail.com (R. Belaroussi), maurice.milgram@upmc.fr (M. Milgram).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.01.076

The tracking can be done in a deterministic way, or use a Kalman-filter or a particle-filter. For instance, in Yin, Zhang, Sun, and Gu (2011a) and Wang, Yang, Xu, and Yu (2009) the Camshift algorithm is combined in a particle-filter approach to track colored objects. We did not investigate the effect of a particle filtering as we considered that the face detection is activated frequently enough to correctly update he skin color model.

In this paper, an efficient scenario for face detection and tracking is proposed, handling multiple faces case, appearance or disappearance of a face anywhere in the scene, and strong illumination variation. Faces are periodically detected during the sequence, using the attentional cascade based on Haar-like filters of Viola and Jones (2001), and the resulting detection are used to compute skin color probability density function (pdf). When a target is already tracked, the face detector is activated every $N = 20$ images of the sequence. Faces are modelled as ellipses based on skin color models: when a face area is less than 100 pixels, it is supposed to have disappear. The corresponding target is released, this way people can enter or get out from any part of the scene, and the camera is not supposed to be fixed. When no target is pursued, the face detector is activated every $N = 2$ images of the sequence, which results in a more time consuming algorithm but handles a new person entrance more rapidly. Skin color is modeled in the Hue-Saturation chromaticity plane of HSV, using the non parametric approach of histograms. Histogram backprojection results in a skin color probability image, which is processed for face localisation by way of connected component segmentation or a coupled Camshift procedure (Bradski, 1998). These two approaches are compared on sequences of 320 × 240 images acquired by a webcam available at i2i (2011).

2. Face detector choice

Four frontal face detectors, Haar-like filter based, are available in OpenCV (2011), and have been well-described by Lienhart, Kuranov, and Pisarevsky (2002). They feature different options that can be summarized as follow:

- input size: 20 × 20 or 24 × 24.
- type of weak classifiers: two or three terminal nodes.
- strong classifiers training algorithm: *Discrete Adaboost* (Freund & Schapire, 1996) or *Gentle Adaboost* (Friedman, Hastie, & Tibshirani, 1998).
- boosted classifiers combination: cascade (Viola & Jones, 2001) or decision tree (Lienhart, Liang, & Kuranov, 2003).

Face detection is done using a sliding window strategy with a scale factor of $f_Q = 1.2$ in this paper. In this section, we describe the image pre-processing used during the face detection step, in order to achieve face tracking in real-time. Then the four detectors performances are compared over a 200 images sequence, in terms of detection rate and speed.

2.1. Image processing for fast and robust face detection

Sequences of 320 × 240 images are taken using a webcam (Yin, Winn, & Essa, 2011b). To perform a fast an reliable face tracking system, the face detection step needs to be rapid and to have a high detection rate. But it also needs to result in the least possible number of false positive, as in our approach a false positive constitute a target to track: a compromise has then to be made. Each image is reduced by a factor two, so that the image processed by the detector is 160 × 120: with this operation face detection is less time consuming, and face candidates are more reliable, as shown by Table 1, but the minimal detectable face size is higher.

Table 1
Performance of the 24 × 24 cascade on the sequence *Antonio* made of 200 images, containing one face per image, with the original image and after reduction.

Image dimension	Face detection performance	Average processing time per image
160 × 120	103 True positives 0 false positive	60 ms
320 × 240	128 true positives 14 false positives	214 ms

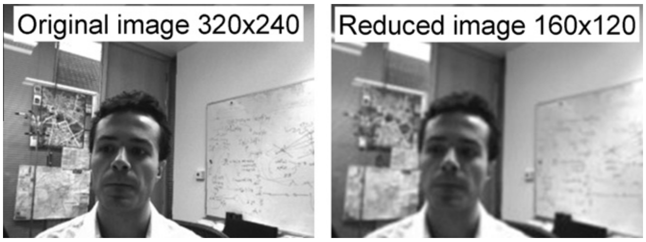

Fig. 1. Example of original (320 × 240) and reduced image (160 × 120).

Table 2
Search window size (width = height) scanning a 160 × 120 at all scales with a magnifying factor $f_Q = 1.2$, for a 24 × 24 cascade. Last line is the corresponding detectable face size in the 320 × 240 original image.

Scale	1	2	3	4	5	6	7	8	9
Sliding window size	24	29	35	41	50	60	72	86	103
Size of detectable faces in 320 × 240 image	48	58	70	82	100	120	144	172	206

Table 3
Search window size (width = height) scanning a 320 × 240 at all scales with a magnifying factor $f_Q = 1.2$, for a 24 × 24 cascade. Last line is the number of sub-windows processed by the cascade at each scale.

Scale	1	2	3	4	5	6	7	8	9	10	11	12	13
Sliding window size	24	29	35	41	50	60	72	86	103	124	149	178	214
Number of sub-windows	9230	9032	8420	8360	7160	4400	2750	1660	966	498	245	94	20

Image reduction is done by a low-pass filtering followed by a sub-sampling (see Fig. 1):

- images are smoothed by convolution with a 5 × 5 Gaussian filter with standard deviation $\sigma = 1.25$. Image smoothing suppresses noises that affect weak classifiers performance, as these classifiers are based on a difference between gray levels in the image. The number of false positive is then lowered.
- sub-sampling multiply by 2 the minimal detectable face size: for example, a 24 × 24 classifier applied on the 160 × 120 image reduced cannot detect faces smaller than 48 × 48 in the 320 × 240 original image, as shown in Table 2. Without the sub-sampling operation, faces smaller than 48 × 48 are searched at 4 more scales (scale 1 to 4 in Table 3 when using a factor $f_Q = 1.2$ between scales). Sliding window sizes refered

in Table 3 correspond to face sizes detectable with a 24 × 24 classifier in the original image. In contrast, performing face detection on the 160 × 120 image requires the investigation of less scale and much less sub-window as referred in Table 2, making this step much less time consuming.
- another consequence of the sub-sampling is that the number of false positive decrease, as there are less non-face sub-images, but also the correct detection rate as less face sub-images are classified.

These points are outlined by applying a 24 × 24 cascade of boosted classifier trained by *Discrete Adaboost* on the 320 × 240 original images and 160 × 120 reduced images of the sequence *Antonio* i2i (2011): see Table 1. It is made of 200 images of a person sitting in front of his monitor with a horizontal back and forth motion (from left to right). Face is tilted at different moments during the sequence (in-plane rotation till ±90°) while the detector results in missed face because it was trained on upright faces database. The subject moves approximatly in a same plan parallel to the camera sensor: its face keep an almost constant size of about 80 × 80 throughout the sequence. Table 1 gives the results of the comparison in terms of accuracy (number of true and false positives) and processing time averaged over the 200 images, using a PIV @2.8 GHz. Processing time of reduced images includes the time of smoothing and sub-sampling. By reducing the image, the number of false positives decrease by 20%, from 128 to 103 over the 200 images of the sequence. On the oher hand, the number of false positive is null due to image reduction, and the average processing time is about a third of that of a 320 × 240. These two points are desirable for our face tracking system:

- it is important to eliminate as much as possible false positive during the face detection step, in order to track consistent targets.
- the detection step has to be fast, that why the detector is only activated every *N* images during the face tracking procedure, the skin color processing being much more rapid than the appearance-based detector. A processing time of 60 ms is acceptable whereas a time of 214 ms visibly slow down the tracking as the retinian persistence is about 50 ms.

Therefore, we decided to keep this image pre-processing step in what follows.

Table 4
Classifiers detection rate over 200 images 160 × 120 of sequence *Antonio*, containing one face per image. For the four detectors, the false positives were eliminated by multiple detection arbitration (these detections were isolated, whereas faces were detected more than two times around a same position).

Input	Classifier	Number of detected faces	Average time per image (ms)
24 × 24	Cascade Adaboost Discret	103	60
20 × 20	Cascade Gentle Adaboost	66	65
20 × 20	Cascade Adaboost Discret 3 terminal nodes	77	61
20 × 20	Arbre Gentle Adaboost	31	103

2.2. Performance comparison of the four detectors

The *Antonio* sequence is also used to compare the four face detectors. Table 4 gives the number of detected faces and the average processing time over the 200 images. The number of false positive is not reported because it was null for the four detectors over this sequence. We can see that the 24 × 24 gives the best results: it is fast and has the highest detection rate. The only drawback of this classifier is that it can detect a face only when it is greater than 48 × 48 in the original image, which means that the subject has to be close enough from the camera. In an application where smaller faces have to be detected, a 20 × 20 would be more adapted. However, the best detection number is 103 out of 200 using the 24 × 24 classifier: this is especially due to images where the subject face is not upright, but also because it is illuminated from the back as shown by Fig. 2. We can see that not only computation time but also performance could be improved using a skin color based tracking algorithm. The next section illustrates these two improvements qualitatively and quantitatively.

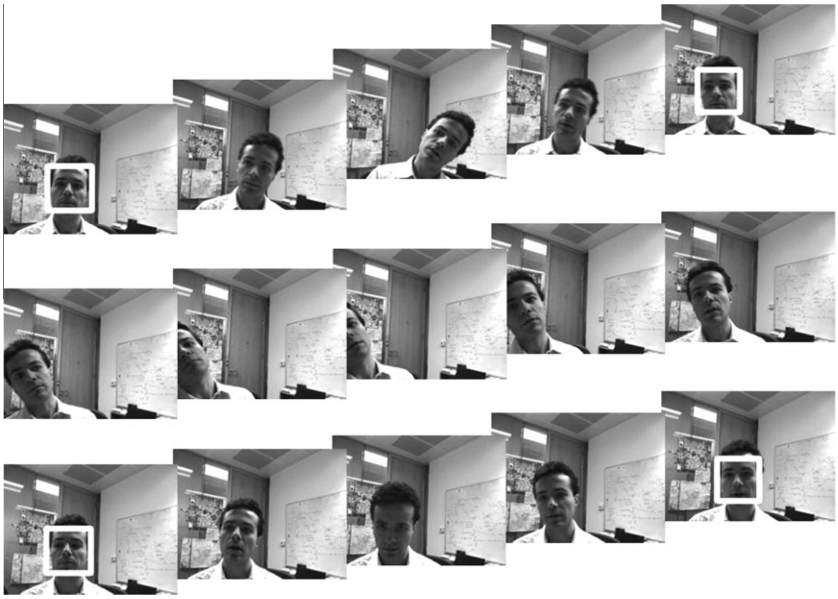

Fig. 2. Example of face detection (white rectangles) results by the 24 × 24 cascade on the *Antonio* sequence.

3. Skin color based tracking: a quantitative comparison of two approaches

3.1. Target model: skin color is modeled at pixel level

Once a face is detected, (on videos results a blue rectangle is drawn) a set of pixels is defined to build the skin color model of the detected person. In order to avoid effects of border pixels in the rectangle localizing the face, where the least reliable colors are, these pixels are taken in an upright ellipse (as faces can only be detected frontally) with a minor and major axis respectively half the width and height of the face bounding rectangle as shown in Fig. 3. It worth noticing that this upright ellipse where the skin color training pixels are taken is quite small compare to the size of the detection rectangle: our experiments leads us to such a choice to be capable of handling challenging background (woods doors, books, wallpaper…) with color close to the face one.

A 32 × 32 bins histogram in the H–S plane is used as a parametric model of the skin color. That is, each of the Hue and Saturation channels are quantified on 32 values. Then each pixel chrominance value (H,S) fall into a bin in the histogram: the 2D histogram count therefore the number of occurrence of each color in the training sample, and is then normalized between 0 and 1 to represent the probabilities of skin colors. The backprojection of this model over the image is a skin color probability image we will call P_{skin}: each pixel of the original image is replaced in P_{skin} by its corresponding probability value in the histogram.

Each target is represented by its state: a face is modelled as an ellipse with a state $\mathbf{s} = (x_t,y_t,\theta,w_t,h_t)$, where (x_t,y_t) are the ellipse center coordinates, θ its angle with the horizontal x-axis, and (w_t,h_t) the minor and major axis of the ellipse. Let's call the ellipse bounding rectangle $\mathbf{R_t}$, centered at ellipse center at time t. In frame at time t, location and spatial extent of the *region of interest* -a limited portion of the image where the face is searched- is updated using face localization $\mathbf{s_{t-1}}$ in frame at $t-1$. The ellipse bounding rectangle $\mathbf{R_{t-1}}$ is magnified by 20% on all its corners as shown by Fig. 4. The region of interest (ROI) is then centered at the face location in the preceding image, and its spatial extent (width and height of the ROI) is bigger enough to handle fast motion of the face. The histogram model of skin color is backprojected only onto the ROI of the image, to compute the skin probability image P_{skin} used to estimate the ellipse state at t.

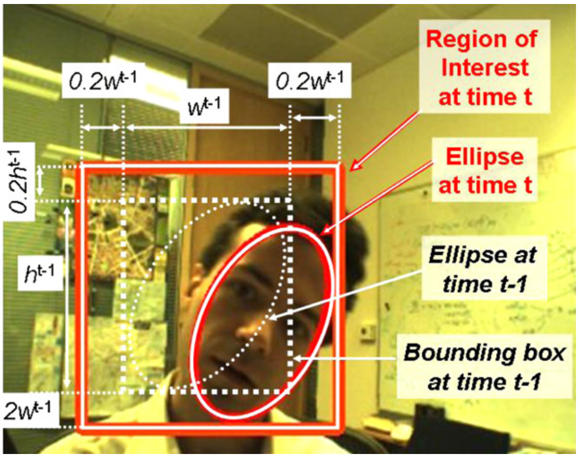

Fig. 4. Target tracking: faces are represented by ellipses tracked in a deterministic way.

3.2. The connected component segmentation approach

Once Pskin, the skin probability image, of the ROI is computed, a simple way to calculate face localization (state of the ellipse bounding the face) can be done in three steps:

- P_{skin} image blurring: dilation and Gaussian filtering.
- Connected component segmentation: threshold and selection of the biggest connected component as face candidate.
- Computation of the best fitting ellipse (in the least square sense) to the contour of the connected component.

Dilation of P_{skin} is necessary due to the relatively bad color quality of images provided by a webcam: without this step, the threshold operation applied later segments the face into two (or more) connected components, one for the upper half of the face (front) and another for the bottom (cheeks,nose and eventually mouth). The structuring element used for dilation is a 3 × 3 square. A Gaussian filter is then applied to smooth the result and fusion the regions of the face. The Gaussian filter size $w_G x h_G$ and standard deviation $\sigma_x x \sigma_y$ depend on face size $w_{t-1} x h_{t-1}$ in the previous frame (i.e. size of the ellipse bounding box at time t-1). This 2D

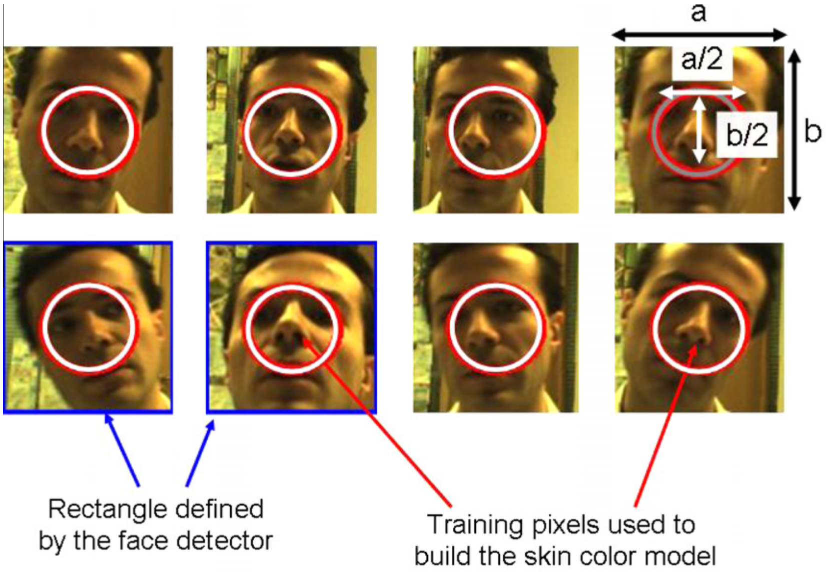
a
a/2
b/2
b
Rectangle defined by the face detector
Training pixels used to build the skin color model

Fig. 3. Training pixels used to build the H-S histogram modelling the skin color are choosen in the middle of face candidate to decrease side effects of non skin pixels.

Gaussian filter is separated into two independant 1D Gaussian filters: column of the image are convolved by the first 1D filter, then rows by the second one:

- Vertical filter: $h_G - h_{t-1}/3$ $\sigma_x = 0.3(h_G/2 - 1) + 0.8$
- Horizontal filter: $w_G = w_{t-1}/3$ $\sigma_y = 0.3(w_G/2 - 1) + 0.8$

Here a Gaussian filter is used for its ability to correctly regularize homogenous regions: a mean filtering would blur the image too strongly and the would alter the face contour. Fig. 6 illustrates the results of these two steps, in the case of two faces of different size, respectively ~120 × 150 pixels and ~40 × 30 pixels. The smoothed image is then binarized by applying a threshold of 50% of smoothed image maximum value: this threshold was found *empirically*. A connected component segmentation is then performed, and the connected component with the biggest area is selected to represent the face.

The contour of that connected component is constituted by a set of 2D points and the best fitting ellipse of the set is computed using the Fitzgibbon, Pilu, and Fisher (1999) algorithm. This method is quite efficient even in the case of distorted contour or unconvex hull as illustrated by Fig. 5. In these examples, the skin color model fails at correctly modelling the face pixels in the dark or overlighted: the resulting connected component has an irregular shape and so is its contour, but the goodness of fit of the ellipse is quite satisfactory.

3.3. The coupled Camshift algorithm: mean shift and Camshift procedures

The Camshift algorithm was introduced by Bradski (1998) for colored object tracking and applied to the face. In this section, mean shift procedure applied to a probability skin image is presented before the Camshift, which is applicable to a still image, and the coupled Camshift for video sequences. Mean shift algorithm can be used on the P_{skin} skin probability image. It starts by initializating a window **W** (scale and location), then while **W** is shifted more than a threshold (1 pixel in our experiments) the following procedure is done:

- gravity center (x_c,y_c) calculation using first moment order of P_{skin}.
- **W** is centered at mean location (x_c,y_c).

Camshift (Continuously Adaptive Mean Shift) algorithm encapsulates the mean shift in a loop for varying window size until convergence. At each iteration, mean shift is applied with a given window size until convergence, then an ellipse is computed based on second order central moments, and window size is updated from the resulting ellipse (See Fig. 7). It can be applied for segmentation of a still image based on skin probability image P_{skin}. Camshift can be viewed as a three step iterative algorithm, starting with an initialized mean shift window **W** (scale and location). While **W** is shifted by more than a threshold (1 pixel in our experiments), do:

- Apply mean shift on skin probability image Pskin, until convergence: store mean location (xc,yc) and value.
- Center **W** at (x_c,y_c) and increment its size by 10 pixels along both direction (+/-5 pixels along width and height) to define a ROI used to compute an ellipse based on inertia moments of skin probability image pixels inside the ROI.
- Compute ellipse major and minor axis (with respective length a and b) projections on x and y direction to define next mean shift window **W**.

The *coupled* Camshift is the Camshift algorithm applied to videos sequence, but instead of applying mean shift again to the same image, it move to the next frame of the sequence. For face tracking, ROI is defined in a deterministic way (see deterministic approach). The coupled Camshift can then be seen as a four steps algorithm to be applied on frame at time t (mean shift window **W** is initialized by its location at time t-1): the three already mentionned step, followed by magnification of **W** rectangle by a 1.4 factor (+/− 20% along width and height) to define face search region (ROI) in the next frame, as illustrated by Fig. 4.

3.4. Comparison of the two tracking approaches

So as explained earlier, faces are detected and tracked by our algorithm, using two different approach. They are compared in this section, as well as the performance of the face detector used alone, over three videos available at i2i (2011). These three sequences are acquired by a webcam, and contain one and only one face per image, except in the sequence *Geoff* where a second face appears in the background but is too small to be detected. These three sequences correspond to different scenario:

- The *Geoff* sequence is made of 48 images. The face of the main subject has a size varying from ~120 × 150 to 140 × 160, and keeps almost upright during the sequence: an example is given in Fig. 6. Face is detected at the first frame, and perfectly tracked by the two approaches. Yet the Camshift is faster with an average of 8 ms/frame, while the connected component processing time is ~12.5 ms, these time include the face detection time. A frame by frame face detection results in 47 correct detection, 1 false positive and 1 missed face, for a computation time of ~60 ms/frame (remember that 160 × 120 reduced images are processed).
- The *Ilkay* sequence is made of 160 images (Yin et al., 2011b). The subject moves from left to right, and its face is harder to detect as it is often in profile view. Face size varies from about 95 × 100 to 120 × 150 pixels. In the beginning of the video, the subject is in a profile view: the 24 × 24 cascade is activated every two frames, and the face is detected at frame number 12 (so the detector has been activated 6 times before the first target initialisation). Then it is correctly tracked by both skin color based approaches. Their performance is then 148 true positives out of 160, and 12 missed. The detection + tracking times are the same as for the *Geoff* sequence, even if Ilkay face size is

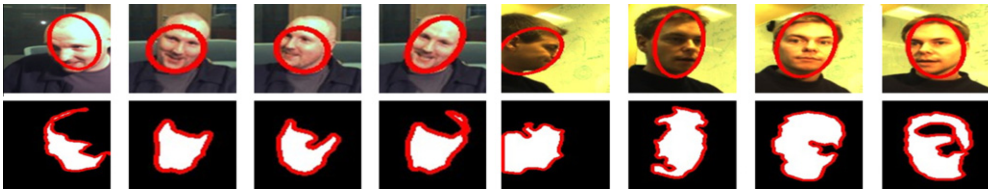

Fig. 5. In these examples, the connected component obtained is in white and its contour is drawn in red. The resulting best fitting ellipse is drawn in red in the original image. Connected component has an irregular shape because of bad illumination condition. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

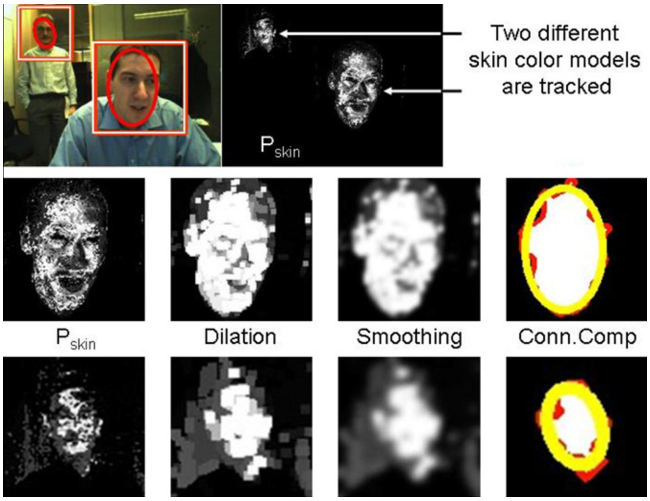

Fig. 6. Post-processing of two ROI with different size: for each target, contour of the biggest connected component is fitted by an ellipse.

much smaller than Geoff's one: this is due to the target initialization computation time, which is quite demanding. Using the face detector alone frame by frame, the average processing time is 62 ms/frame, and the performance is 132 true positives out of 160, 28 faces are missed, and 1 false positive.

- The *Jamie* sequence is made of 85 frames, and is the most challenging, because of the extreme lighting conditions encountered. Background is yellow close to skin color, and the subject is lightened by behind with a neon source. We can see the shadow on the face in the first two examples of Fig. 8. *Jamie* video contains one face by image: target initialization occures at frame number 3, after that, skin color model is updated at three different instant. At frame 31, denoted LIGHT++ in Fig. 8, an additional light is switched on, then the face is in over-exposure condition: luminance of three quarter of the face is over 255, as we can see on the P_{skin} reported on the top right of the frame. The connected component segmentation results in a small false positive around the mouth or the ear, and later on in a too small region: the face is supposed to have disappeared from the scene. The face detector is therefore activated every two frames, slowing down the algorithm until the face appears upright again (20 frames later). The coupled Camshift behaves well when P_{skin} is full of whole, whatever the face pose is. Detection + face tracking by connected component segmentation takes about 17.5 ms/frame, with 43 true positives, 42 missed faces and 20 false positives. Meanwhile, Camshift is very efficient with 8.5 ms/frame, and 83 true positives and 2 missed faces. A frame by frame face detection takes about 59 ms/frame, with 61 true positives and 24 missed.

These results are summarized in Table 5, averaged over the three sequences.

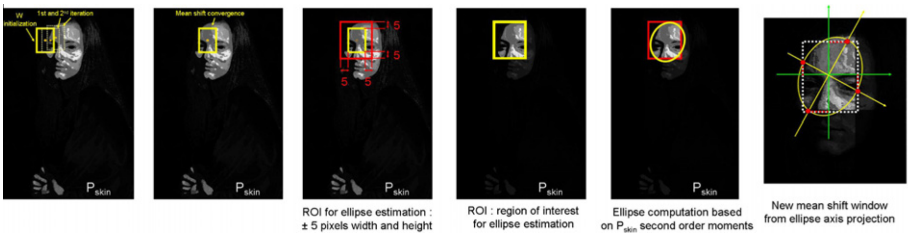

Fig. 7. One pass of the Camshift procedure over a still image.

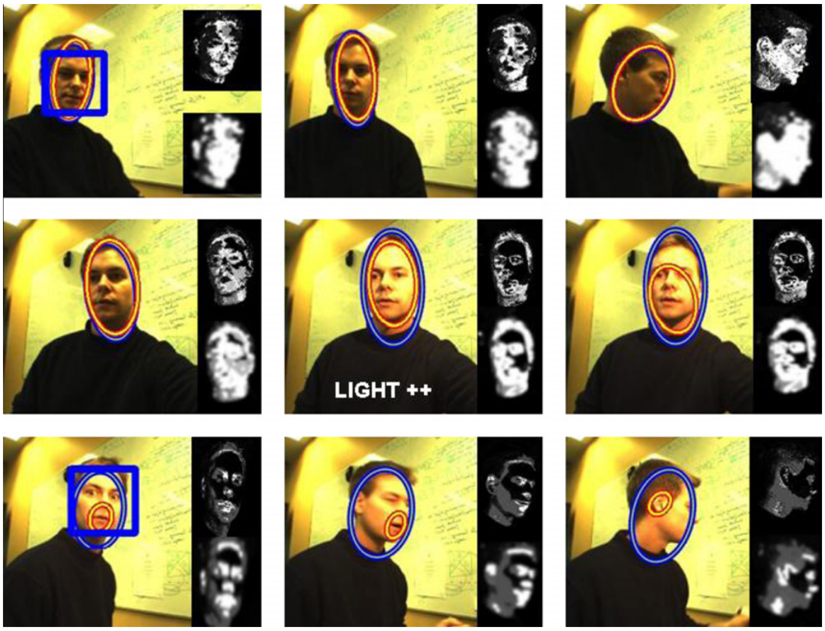

Fig. 8. Examples of face detection (blue rectangles) and tracking in *Jamie* sequence: blue ellipses are the result of Camshift, red and yellow ellipses are the connected component segmentation result. Gray scale stamps on the right illustrates the skin probability image P_{skin} and its smoothing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Performance averaged over the 293 images of the three sequences, with the average time spent in face detection and in skin color tracking.

Tracking approach	Performance	Processing time by frame	
		Detect.	Track.
Face detection frame by frame	240 true positives 53 missed 2 false positives	60 ms	
Connected Component	239 true positives 54 missed 20 false positives	13.5 ms 8 ms	 5.5 ms
Coupled Camshift	279 true positives 2 missed 0 false positive	8.5 ms 5.5 ms	 3 ms

4. Conclusion

In this paper, we proposed an efficient approach for face detection and tracking. A state of the art face detector has been selected amongst four classifiers, available in OpenCV,that were compared in terms of accuracy and speed. The 24 × 24 cascade of boosted classifier has been used for target initialization, and faces were tracked based on skin color modelled using a non-parametric approach. Two skin color based tracking approaches were compared: connected component segmentation and coupled Camshift. This last technique has proven to be the fastest and the more accurate, enabling a detection and tracking automated system than runs in less than 9 ms/frame on a PIV @2.8 GHz.

References

Belaroussi, R., Prevost, L., & Milgram, M. (2005). Combining model-based classifiers for face localization. In *Proceedings of the ninth IAPR conference on machine vision applications* (pp. 290–293).

Bradski, G. (1998). Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal, Q2*(15). URL citeseer.ist.psu.edu/bradski98computer.html.

Fitzgibbon, A. W., Pilu, M., & Fisher, R. B. (1999). Direct least-squares fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(5), 476–480.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *International conference on machine learning* (pp. 148–156).

Friedman, J., Hastie, T., & Tibshirani, R. (1998). Additive logistic regression: A statistical view of boosting. Technical report, Departement of Statistics, Stanford University.

Hu, M., Worrall, S., Sadka, A., & Kondoz, A. M. (2004). Automatic scalable face model design for 2d model-based video coding. *Signal Processing: Image Communication, 19*(5), 421–436.

i2i, 2011. Microsoft research cambridge. i2i: 3D Visual Communication. URL research.microsoft.com/vision/cambridge/i2i/DSWeb.htm.

Jones, M., & Viola, P. (2003). Fast multi-view face detection. Technical report, Mitsubishi Electric Research Laboratories., present demonstration Computer Vision and Pattern Recognition.

Kim, J.-B., Kee, S.-C., & Kim, J.-Y. (2005). Fast detection of multi-view face and eye based on cascaded classifier. In *Proceedings of the ninth IAPR conference on machine vision applications* (pp. 116–119).

Li, S. Z., & Zhang, Z. (2004). Floatboost learning and statistical face detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 26*(9), 1112–1123.

Li, Y., Goshtasby, A., & Garcia, O. (2000). Detecting and tracking human faces in videos. In *Proceedings of the 15th international conference on pattern recognition* (Vol. 1, pp. 807–810).

Lienhart, R., Kuranov, A., & Pisarevsky, V. (2002). Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Technical report, Microprocessor Research Lab, Intel Labs.

Lienhart, R., Liang, L., & Kuranov, A. (2003). A detector tree of boosted classifiers for real-time object detection and tracking. In *IEEE international conference on multimedia & expo*.

McKenna, S. J., Gong, S., & Raja, Y. (1998). Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition, 31*(12), 1883–1892.

Swain, M. J., & Ballard, D. B. (1991). Color indexing. *International Journal of Computer Vision, 7*(1), 11–32.

OpenCV. (2011). Open source computer vision library. URL http://sourceforge.net/projects/opencvlibrary/.

Phung, S., Bouzerdoum, A., & Chai, D. (2005). Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(1), 148–154.

Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(1), 23–38.

Schneiderman, H., & Kanade, T. (2000). A statistical model for 3d object detection applied to faces and cars. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (Vol. 1. pp. 746–751).

Schumeyer, R., & Barner, K. (1998). Color-based classifier for region identification in video. In *Proceedings of the SPIE visual communications image processing* (Vol. 3309. pp. 189–200).

Schwerdt, K., & Crowley, J. L. (2000). Robust face tracking using color. *Proceedings of the fourth IEEE international conference on automatic face and gesture recognition* (pp. 90). Washington, DC, USA: IEEE Computer Society.

Seguier, R. (2004). A very fast adaptive face detection system. In *International conference on visualization, imaging, and image processing*.

Soriano, M., Martinkauppi, B., Huovinen, S., & Laaksonen, M. (2003). Adaptive skin color modeling using the skin locus for selecting training pixels. *Pattern Recognition, 36*(3), 681–690.

Sung, K.-K., & Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(1), 39–51.

Vezhnevets, V., Sazonov, V., & Andreeva, A. (2003). A survey on pixel-based skin color detection techniques. In *Proceedings of Graphicon-2003, Moscow, Russia* (pp. 85–92).

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (Vol. 1. pp. 511–518).

Wang, Z., Yang, X., Xu, Y., & Yu, S. (2009). Camshift guided particle filter for visual tracking. *Pattern Recognition Letters, 30*(4), 407–413.

Wu, B., Ai, H., Huang, C., & Lao, S. (2004). Fast rotation invariant multi-view face detection based on real adaboost. In *Proceedings of the sixth IEEE international conference on automatic face and gesture recognition* (pp. 79–84).

Yang, J., & Waibel, A. (1996). A real-time face tracker. *Proceedings of the 3rd IEEE workshop on applications of computer vision* (pp. 142). Washington, DC, USA: IEEE Computer Society.

Yang, M., & Ahuja, N. (1999). Gaussian mixture model for human skin color and its application in image and video databases. In *Proceedings of the SPIE conference on storage and retrieval for image and video databases* (pp. 458–466).

Yin, M., Zhang, J., Sun, H., & Gu, W. (2011a). Multi-cue-based camshift guided particle filter tracking. *Expert Systems with Applications, 38*(5), 6313–6318.

Yin, P., Winn, J., & Essa, I. (2011b). Bilayer segmentation of webcam videos using tree-based classifiers. *Trans. Pattern Analysis and Machine Intelligence, 33*(1), 30–42.

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

Zhu, X., Yang, J., & Waibel, A. (2000). Segmenting hands of arbitrary color. *Proceedings of the Fourth IEEE international conference on automatic face and gesture recognition* (pp. 446–453). Washington, DC, USA: IEEE Computer Society.