ON-LINE FACE TRACKING UNDER LARGE LIGHTING CONDITION VARIATIONS USING INCREMENTAL LEARNING

Lyes Hamoudi, Khaled Boukharouba, Jacques Boonaert
and Stéphane Lecoeuche
Ecole des Mines de Douai, Département Informatique et Automatique
941 Rue Charles Bourseul, 59508 Douai, France

Keywords: Face detection and tracking, Colour and texture segmentation, Classification of non-stationary data, SVM classification.

Abstract: To be efficient outdoors, automated video surveillance systems should recognize and monitor humans activities under various amounts of light. In this paper, we present a human face tracking system that is based on the classification of the skin pixels using colour and texture properties. The originality of this work concerns the use of a specific dynamical classifier. An incremental svm algorithm equipped with dynamic learning and unlearning rules, is designed to track the variation of the skin-pixels distribution. This adaptive skin classification system is able to detect and track a face in large lighting condition variations.

1 INTRODUCTION

A fundamental and challenging problem in computer vision is the detection and tracking of faces and facial features in video sequences. Face detection area has applications in various fields, like Video surveillance, Security control systems, Human-computer interaction (HCI), Videophony and Videogames. Many researchers proposed different methods addressing the problem of face detection, and there are several possibilities to classify these methods. In their survey, (Yang et al., 2002) classified different techniques used in face detection as Knowledge-based methods, Feature-based methods, Template matching methods and Appearance-based methods. Among feature-based face detection methods, the ones using skin colour segmentation have gained strong popularity. They are orientation invariant and computationally inexpensive to process, since colour is a low-level property (Martinkauppi, 2002). It is therefore suitable for real-time systems.

A problem with skin colour segmentation arises under varying lighting conditions. The same skin area appears as two different colours under two different lighting conditions (Sigal et al., 2004). Several approaches have been proposed to use skin colour in varying lighting conditions. (McKenna et al., 1999) presented an adaptive colour mixture model to track faces under varying illumination conditions. (Strring et al., 1999) estimated a reflectance model of the skin, using knowledge about the camera parameters and the light source spectrum. They estimated the location of the skin colour area in the chromaticity plane for different light sources. (Soriano et al., 2000) transformed the RGB pixel map to Normalized Colour Coordinates (NCC) allowing a pixel brightness dependence reduction. In their work, a chromaticity histogram of some manually selected skin pixels is used as an initial, non-parametric colour model. (Sigal et al., 2004) described an approach for real-time skin segmentation in video sequences, which enables segmentation despite wide variation in illumination during tracking. They used an explicit second order Markov model to predict evolution of the skin-colour (HSV) histogram over time. (Chow et al., 2006) presented an algorithm where skin-coloured pixels are identified using a region-based approach. They proposed a colour compensation scheme to balance extreme lighting conditions, and the distributions of the skin-colour components under various illuminations are modelled by means of the maximum-likelihood method.

(La Cascia et al., 2000) proposed an algorithm for 3D head tracking that uses a texture mapped 3D rigid surface model for the head. They use a method that employs an orthogonal illumination basis that is pre-computed off-line over a training set of face images collected under varying illumination conditions. These proposed methods for handle the variation of lighting conditions are based on a modelling, estimation or a prediction of the skin colour, or a colour compensation scheme. For most of them, large sets

Hamoudi L., Boukharouba K., Boonaert J. and Lecoeuche S. (2009).
ON-LINE FACE TRACKING UNDER LARGE LIGHTING CONDITION VARIATIONS USING INCREMENTAL LEARNING.
In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, pages 636-643
DOI: 10.5220/0001806106360643

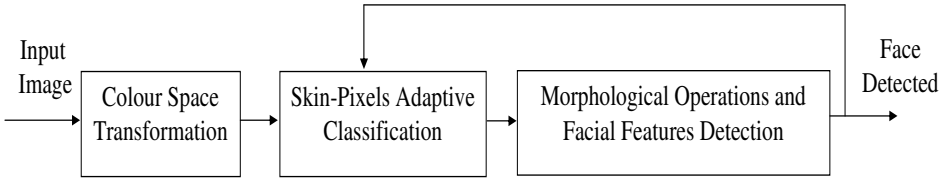

Figure 1: Face detection scheme.

of images of face under varying illumination conditions are required in the learning stage. In this paper, we propose a face tracking method that is able to handle large lighting condition variations, without using large training sets of image samples. The method uses an incremental svm classification algorithm equipped with dynamic learning and unlearning rules (Cauwenberghs and Poggio, 2000). It is designed to track the variation of skin-pixels distribution in the feature space over time. This property allows an on-line adaptation of the skin-pixel cluster discriminate function.

We begin this paper by introducing the method scheme, in the second section. We develop three stages (colour space transformation, skin-pixels adaptive classification, morphological operations and facial features detection). Skin-pixels adaptive classification is the key feature of our face tracking method. The paper ends with an experimental evaluation of the system.

2 METHOD PRESENTATION

Our face tracking system is composed of three steps (Figure 1). It begins with the transformation of the input RGB image into *THS* format (*Texture, Hue* and *Saturation*). This new format avoids intensity component and so, it is less sensitive to lighting variations. At the second step, a dynamic classification of each pixel in the image as a skin-pixel or a non-skin-pixel is done. This classification allows tracking the skin-pixels cluster over time. Finally, the system identifies different skin regions in the skin detected image by using morphological operations and geometrical analysis. The last stage is designed to decide whether each of the skin regions identified is a face or not by looking for features as eyes and mouth and spatial relation between these features. For each frame, the pixels that are part of the region recognised as a face are incrementally added into the skin-pixels model using specific learning procedures. To achieve the skin-pixels model adaptation, the pixels learned at the latest frames will be decrementally removed using unlearning procedures.

2.1 Colour Space Transformation

The Colour space transformation is based on the (Forsyth and Fleck, 1999) algorithm. The original colour image is in RGB format. The R, G, and B values are transformed into log-opponent values I, Rg, and By, and from these values *Texture*, *Hue*, and *Saturation* are computed (Forsyth and Fleck, 1999).

Human faces have a distinct texture that can be used to separate them from different objects (Cula et al., 2005), and skin in images tends to have very smooth texture. A *Texture* map is used to find regions of low texture information. To generate this *Texture* map, the original image I is filtered by a median filter Ψ, the filtered image is subtracted from I, and the absolute value of the difference is filtered again by Ψ. *Hue* and *Saturation* are used to select those regions whose colour matches that of skin. They are simply the direction and magnitude of the vector (Rg, By), and are calculated as:

$$\begin{aligned} Texture &= \Psi(|I-\Psi(I)|) \\ Hue &= \arctan^2(Rg,By) \\ Saturation &= \sqrt{(Rg^2+By^2)} \end{aligned} \quad (1)$$

Figure 2.a represents the RGB image, and Figures 2.b, 2.c and 2.d represent respectively the resulting *Texture* map, *Hue* and *Saturation* components.

2.2 Skin Pixels Marking

With *Texture*, *Hue*, and *Saturation* components, regions of skin can be extracted using a classification task. For the skin-pixels classification, a simple and commonly used method defines skin to have a certain range or values in some coordinates of a colour space. This can easily be implemented as a look-up table or as threshold values (Chai and Ngan, 1998). With empirically chosen thresholds [Tex_L, Tex_H], [Hue_L, Hue_H] and [Sat_L, Sat_H], a pixel is classified as being a skin-pixel if its values *THS* fall within the ranges (i.e. Tex_L $< T <$ Tex_H, and Hue_L $< H <$ Hue_H, and Sat_L $< S <$ Sat_H). Thus, if a pixel is classified as a skin-pixel it is marked in a binary skin map array where 1 corresponds to the coordinates being a

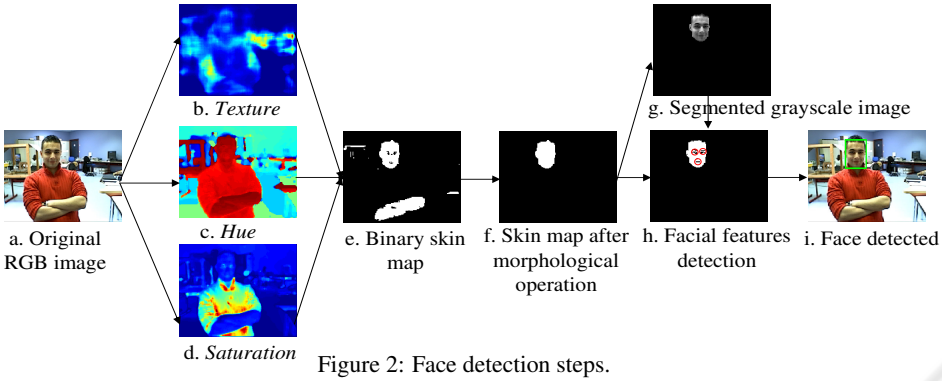

Figure 2: Face detection steps.

skin pixel in the original image and 0 corresponds to a non-skin pixel. The skin map array can be considered as a black and white binary image with skin regions appearing as white, and the non-skin regions as black, see Figure 2.e.

The method using thresholds works fairly well, and tolerates some illumination variations (Chai and Ngan, 1998). Nevertheless, in large lighting variation, it proves defective, since the distribution of the skin-pixels in the feature space should be significantly changed over time. Therefore, the thresholds need to be updated. To avoid this, in the section 3, a classification algorithm is presented. Using this technique, the decision model will be updated according to the non-stationary of data that characterise the skin pixels cluster. Before presenting the skin-pixels adaptive classification, the next section presents the last stage of our face detection scheme.

2.3 Morphological Operations and Facial Features Extraction

The binary skin map regions are processed by morphological operations for delete noise, close holes and separate regions. Since a face has an elliptical shape with usually a vertical orientation, we delete the regions that have not an elliptical shape, as well as those that have a horizontal orientation that could correspond to an arm or to an object with colour and texture similar to human skin. The remaining regions in the skin map represent the face candidates, see Figure 2.f. Finally, these candidates are verified by searching for facial features inside the regions. The technique relies on searching for darker parts (holes) in the skin regions, so that these holes would correspond to eyes and mouth. The region that contains holes with triangular spatial relation is validated as being a face; see Figure 2.g and 2.h and 2.i. The other regions are discarded.

3 SKIN-PIXELS ADAPTIVE CLASSIFICATION

This section details the algorithm used to classify skin pixels under large lighting variations. The goal is to label the pixels into skin-pixels and non-skin ones using an update decision model. For that, we use an incremental svm classifier equipped with learning and unlearning rules (Cauwenberghs and Poggio, 2000) that will allow the tracking of the cluster evolution due to lighting condition variations. Figure 3 illustrates the need of cluster adaption by drawing a decision function at time t, and at time $t+N$. C is the cluster or the model of skin-pixels and f_t its temporal boundary decision function, that we will simply call the boundary. At each frame (time=t) of the video sequence, each pixel $x_i(T_i, H_i, S_i) \in \mathbb{R}^3$ will be classified as a skin-pixel if $f_t(x) \geq 0$. So,

$$\begin{array}{llll} \text{if} & f_t(x) \geq 0, & \text{then} & x \in C \\ \text{if} & f_t(x) < 0, & \text{then} & x \notin C \end{array} \quad (2)$$

f_t is defined in *THS* feature space (*THS* space) by :

$$f_t(x) = \sum_i^d \alpha_i K(x_i, x) + \rho \quad (3)$$

ρ is the offset of the function and $K(\bullet, \bullet)$ is the RBF kernel. d is the number of skin-pixels. The weights α_i are the Lagrange multipliers and they are obtained by minimizing a convex quadratic objective function (Vapnik, 1995):

$$\min_{0 \leq \alpha \leq C} : W = \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j) + \sum_i \alpha_i \rho - \rho \quad (4)$$

The boundary function f_t is adapted by adding the skin-pixels of the new frames and by removing those of the old frames. For example, on the Figure 3, the triangular dots represent the skin-pixels at time t, and the circular dots represent the skin-pixels at

time $t+N$. The adaptation of the decision function will be done by incrementally adding the full circular dots, and by incrementally removing the full triangular dots.

The key is to add each new pixel to the solution while always retaining the Karush Kuhn Tucker (KKT) conditions satisfied on all previously seen pixels. The first order conditions on the gradient of W lead to the KKT conditions:

$$\begin{aligned} g_i = \frac{\partial W}{\partial \alpha_i} &= \sum_{j=1}^{s} \alpha_j K(x_i,x_j) + \rho \\ &= f(x_i) \quad \left\{ \begin{array}{ll} >0; & \alpha_i = 0 \\ =0; & 0<\alpha_i<a, a=cste \\ <0; & \alpha_i = a \end{array} \right\} \qquad (5) \\ \frac{\partial W}{\partial \rho} &= \sum_{j=1}^{s} \alpha_j - 1 = 0 \end{aligned}$$

where s is the number of support vectors.

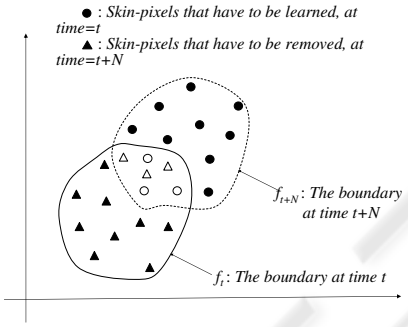

Figure 3: Decision function (the boundary) adaptation according to the skin-pixels cluster evolution.

This classification divides all the pixels of the image into 3 sets:

- The set D consists of the inside vectors, which are situated within the boundary $(\forall x_i \in D, g_i > 0)$.
- The set S consists of the support vectorhis, which are situated on the boundary $(\forall x_j \in S, g_j = 0)$. $j = 1,\ldots,s$.
- The set U consists of the uncertain vectors, which are situated outside of the boundary $(\forall x_u \in U, g_u < 0)$.

C is the skin-pixels cluster, so $D \cup S = C$. A pixel classified in D or S is immediately classified as a skin-pixel. In addition, for every pixel x_i classified in C, the value g_i is stored in a set of associated values G. When a pixel is classified in U, two cases should be considered. In most of the cases, this pixel is not a skin pixel, his attributes being too far from the skin class model. But for some cases, this pixel could be considered as a skin pixel where his colour attributes have changed due to lighting variation and then should be used to update the skin cluster model. A similarity measure is then required to select these uncertain pixels.

3.1 Similarity Measure

In the principle of updating the skin-pixels model over time, whenever a new or a candidate pixel x_c is classified in U, it is assumed that if is fairly close to the boundary, it could correspond to a skin pixel, having undergone a change of lighting. Then, we calculate the distance of this pixel from the boundary. If this distance is too large, the pixel will be discarded. But if it is small enough, it will be added to S and the boundary function will be adjusted until x_c is on the boundary, so $g_c = 0$. For that, we introduce a new measure of similarity. We calculate in the Hilbert feature space Γ (see Figure 4) the angle between the candidate pixel x_c and every support vector.
The dot product between x_c and x_j is expressed as:
$\langle \phi(x_c), \phi(x_j) \rangle = \|\phi(x_c)\|_\Gamma \left\|\phi(x_j)\right\|_\Gamma \cos(\phi(x_c), \phi(x_j))$.

Therefore using RBF kernel, the smallest angle θ_{nst} is expressed as:

$$\theta_{nst} = \arg\min_j \left\{ \cos^{-1}(K(x_c,x_j)), x_j \in \{S\} \right\} \qquad (6)$$

We compare the smallest angle with a threshold θ_{sim}. If $\theta_{nst} \leq \theta_{sim}$, x_c will be added to S and the boundary function will adjust, else it will be discarded.

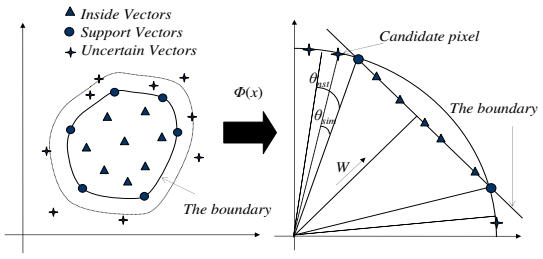

Figure 4: Illustration of data projection in the Hilbert space using Gaussian kernel.

3.2 Incremental Learning

When x_c is added to the set S, the parameters of the skin cluster boundary function is updated iteratively. At every iteration, $f_t(x)$ is adapted until $g_c = 0$. z is the set of the parameters $\{\rho, \alpha_j\}$. These parameters

change to keep their KKT conditions satisfied. For that, those conditions are expressed differentially as:

$$\begin{aligned} \Delta g_i &= K(x_i,x_c)\Delta\alpha + \sum_{j=1}^{s} \Delta\alpha_j K(x_i,x_j) + \Delta\rho \\ 0 &= \Delta\alpha + \sum_{j=1}^{s} \Delta\alpha_j,\ j=1,\ldots,s, i=1,\ldots,d \end{aligned} \tag{7}$$

Since $g_j = 0$ for every support vector, the changes in weights must satisfy

$$\underbrace{\begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K(x_1,x_1) & \cdots & K(x_1,x_s) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_s,x_1) & \cdots & K(x_s,x_s) \end{bmatrix}}_{Jacobian\ Q} \cdot \underbrace{\begin{bmatrix} \Delta\rho \\ \Delta\alpha_1 \\ \vdots \\ \Delta\alpha_s \end{bmatrix}}_{Delta} = - \underbrace{\begin{bmatrix} 1 \\ K(x_1,x_c) \\ \vdots \\ K(x_s,x_c) \end{bmatrix}}_{h} \cdot \Delta\alpha$$

So, $Delta = -\underbrace{Q^{-1}}_{R} \times h \times \Delta\alpha$.

Thus, should be defined:

$$\begin{aligned} \Delta\rho &= \beta_0.\Delta\alpha \\ \Delta\alpha_j &= \beta_j.\Delta\alpha, \forall x_j \in S \end{aligned} \tag{8}$$

with weights sensitivities given by

$$\begin{bmatrix} \beta_0 & \beta_1 & \cdots & \beta_s \end{bmatrix} = -R \cdot h \tag{9}$$

Where $R = Q^{-1}$, and $\beta = 0$ for all x outside S. By this way, the values of the incrementation steps of all the parameters are calculated.

Parameters Update. The associated value g_c and the weight α_c of the pixel x_c added to S, will be added to the set G and the set A_i as: $G^{s+1} \leftarrow G^s \cup \{g_c\}$ and $A_i^{s+1} \leftarrow A_i^s \cup \{\alpha_c\}$. The matrix R (Q) will be updated by adding a line and a column corresponding to the new pixel x_c.

Gradient set Update. When x_c is added to S, according to the update of the boundary function, all the elements of G should be modified. Then:

$$\forall x_i \in D, \quad \Delta g_i = \gamma_i \Delta\alpha, \quad i = 1,\ldots,d \tag{10}$$

where γ_i is defined as:

$$\gamma_i = K(x_i,x_c) + \sum_j K(x_i,x_j).\beta_j + \beta_0,$$
$$i = 1,\ldots,d, j = 1,\ldots,s$$

When x_c is added to D, α_c is equal to 0 and only G will be incremented: $G^{s+1} \leftarrow G^s \cup \{g_c\}$.

Remark 1. During the adjustment, a support vector x_j that was on the boundary could be found inside the boundary, (i.e. by the incrementation procedure, to keep the KKT conditions satisfied, α_c could end up equal to zero), in this case x_j will be eliminated from S and put in D, and all ρ, α_j and R will be updated.

Incremental Learning Algorithm. To conclude, the incremental learning procedure for a candidate pixel x_c is defined as:

Initialize α_c to zero.
If $g_c > 0$, add x_c to D, update G, terminate.
If $g_c = 0$, add x_c to S, update α_j and ρ (Eq.8), R and G, terminate.
If $g_c < 0$, add x_c to U, calculate the angle θ_{nst}.
  If $\theta_{nst} < \theta_{sim}$ do
    add x_c to S
    While $g_c < 0$ do
      $\alpha_c = \alpha_c + \Delta\alpha$
      Calculate $\Delta\rho$ (Eq.8), $\rho = \rho + \Delta\rho$
      for each $x_j \in S$,
      calculate $\Delta\alpha_j$ (Eq.8), $\alpha_j = \alpha_j + \Delta\alpha_j$
      for each $x_i \in C$,
      calculate Δg_i (Eq.10), $g_i = g_i + \Delta g_i$
      Check if a support vector (or several) passes inside the boundary ($\alpha_j \leq 0$). If true, delete x_j from S and add it to D, and update all the parameters.
Repeat as necessary (until $g_c = 0$).

Remark 2. The initial learning is done using the first frames of the video sequence. On these frames, we apply a face detection using the thresholds method (section 2.2). We obtained several series of thresholds by collecting several skin-pixels models (in *THS* space) using different video sequences with various people under different lighting conditions. On each of the first frames, we apply the thresholds method using one series of thresholds at time. One of the series leads to obtain the better face detection among the others. So, once the face detected, the pixels recognized as being skin pixels are presented to the classifier one pixel by one, without using the test of similarity, and all will be granted to C, constructing by this way the initial boundary, that will be used for the tracking.

3.3 Decremental Learning

The unlearning procedure complements the learning procedure to allows the system to track the cluster C over lighting condition variations, by forgetting (removing from C) the previous learned data that correspond to skin-pixels at the initial learning stage, and after, those on the oldest frames. When the system process the N^{th} frame, the skin-pixels learned from the $(N-m)^{th}$ frame correspond to obsolete information and then should be forgotten. When a pixel x_j is removed from S, g_j will be removed from G, and z

will be adapted decrementally and the boundary will move until x_j is out ($\alpha_i \leq 0$). The matrix R is updated by deleting from the matrix Q the column j+1 and the line $j+1$ (corresponding to x_j that has been removed). When a pixel x_i is removed from D, only g_i is removed from G.

Decremental Unlearning Algorithm. When removing the pixel x_r from C, the parameters $\{\alpha^{s-1}, \rho^{s-1}\}$ are expressed in terms of the parameters $\{\alpha^s, \rho^s\}$, the matrix R, and x_r as:

If $g_r > 0$, ($x_r \in D$) remove x_r from C, $G \leftarrow G - \{g_r\}$, terminate.
If $g_r = 0$, remove x_r from S (and thus from C),
While $\alpha_r > 0$, do $\alpha_r = \alpha_r - \Delta\alpha$
Calculate $\Delta\rho$ (Eq.8), $\rho = \rho - \Delta\rho$
for each $x_j \in S$,
calculate $\Delta\alpha_j$ (Eq.8), $\alpha_j = \alpha_j - \Delta\alpha_j$
for each $x_i \in C$,
calculate Δg_i (Eq.10), $g_i = g_i - \Delta g_i$
Check if an inside vector $x_i \in D$ (or several) passes outside the boundary ($g_i \leq 0$). If true, interrupt the decremental unlearning, and apply the incremental learning on x_i.
Return to the decremental unlearning procedure.
Repeat as necessary (until $\alpha_r = 0$).

4 EXPERIMENTS

At first, we performed experiments on video sequences collected in our laboratory by a Philips SPC900NC/00 web-cam (settings frame rate = 30fps, image size 160x120 pixels). Each sequence is 600 frames long (20 seconds). The camera was mounted on a laptop and volunteers were asked to sit down in front of the laptop and perform free head motion while we greatly vary the lighting, passing through a very dark to a very enlightened state. We first applied the tracking method using thresholds. This method works quite well under constant lighting, but fails when the lighting varies significantly. We then applied the method using the incremental classification. We fixed the kernel parameter $\sigma = 5$ and the threshold angle $\theta_{sim} = 1rad$ after applying several experimentations. The unlearning procedure is started at the 5^{th} frame, i.e. when the system process the N^{th} frame of the video sequence, it unlearns the pixels learned from the $(N-5)^{th}$ frame. This value proved to be efficient for a reliable on-line tracking of skin-pixels cluster. The obtained results were very encouraging, since the face was accurately tracked on all the video sequences, except in the frames were the face was in profile (because of the conditions to find the two eyes and the mouth).

Secondly, we performed experiments on the set of sequences collected and used by (La Cascia et al., 2000). The set consists of 27 sequences (nine sequences for each of three subjects) taken under time varying illumination and where the subjects perform free head motion. The time varying illumination has a uniform component and a sinusoidal directional component. It should be noted that the time varying illumination is done in a non-linear manner, by darkening the scene and specially the right side, making the right side of the face extremely dark. In addition, the free head motion is performed such that the face is never completely in profile. All the sequences are 200 frames long (approximatively seven seconds), and were taken such that the first frame is not always at the maximum of the illumination. The video signal was digitized at 30 frames per second at a resolution of 320x240 non-interleaved using the standard SGI O2 video input hardware and then saved as Quicktime movies (M-JPEG compressed). All of these sequences are available on-line: http://www.cs.bu.edu/groups/ivc/HeadTracking/, (and are the only available among those used in the articles cited in the introduction). Figure 5 shows examples of images of the three subjects from the video sequences, showing time varying illumination and free head motion.

Figure 6 shows the mean values of the manually extracted skin-pixels on the 200 frames of a video sequence, in the RGB colour space and in the *THS* colour space. We can see that while there is a great variance in the RGB, the *THS* is less sensitive to lighting variation. As defined, we see that the *Texture* and the *Hue* component have smooth values and are quite constant through lighting variation. In addition, the *Saturation* component is more dependent on great lighting variation. In this case, it becomes clear that the threshold method cannot obtain good results. Thus, an adaptive classification method is needed to track the skin-pixels cluster through the time varying illumination. We cannot objectively compare our results to those of (La Cascia et al., 2000), because their system uses a texture mapped 3D rigid surface model for the head. In addition, the output of their system is the 3D head parameters and a 2D dynamic texture map image. We just note that a version of the tracker that used a planar model was unable to track the whole sequence without losing track.

To evaluate our tracker, we first linked the nine

sequences of each subject, obtaining three sequences of 1800 frames. After, as ground truth, we manually surrounded the face (black bounding box on Figure 5 on all the frames of each sequence, and we calculate the horizontal and vertical coordinates X and Y of the center of the bounding box surrounding the face. Then, we applied our tracking algorithm on each sequence, aiming to detect and track the face skin-pixels. We also calculate the horizontal and vertical coordinates X and Y of the center of the bounding box surrounding each detected face (white bounding box on Figure 5). Figure 7 shows the curves of the evolution of the coordinates X and Y of the center of the ground truth faces, superposed to those of the detected faces. The pseudo sinusoidal demeanor of the curves is due to the free head motion of the subjects, and the abrupt transition corresponds to where we linked the nine sequences. We can see that the ground truth bounding box and the detected bounding box are practically overlapped, proving that all the face regions were detected. The centers of the bounding box diverge only in the case where a face side is heavily dark, so this part of the face was not considered as containing skin by the tracker.

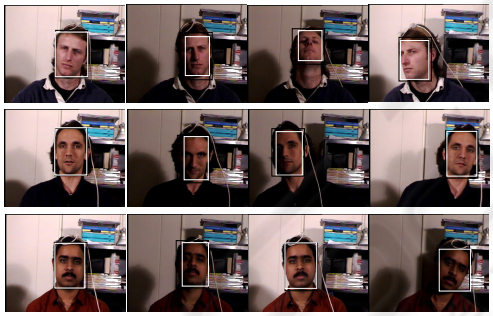

Figure 5: Example of images of the three subjects from the video sequences, showing time varying illumination and free head motion. With face detection results (white bounding box) and ground truth (black bounding box).

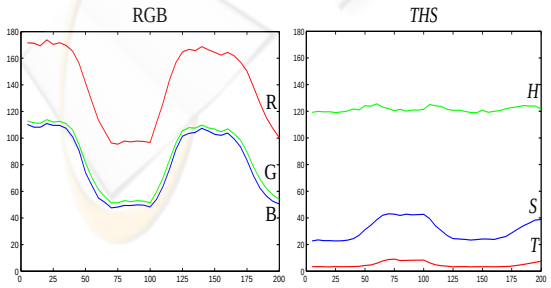

Figure 6: Mean values of manually extracted skin-pixels on 200 frames of a video sequence, in the RGB colour space and in the *THS* colour space.

5 DISCUSSION

The results obtained by our algorithm are very encouraging. Nevertheless, it can still be improved on several fronts. For example, we plan to develop an initialization method without using a pre-computed off-line set of skin-pixels models. In addition, we plan to develop a detection validation method that do not need that the face is in front of the camera. Furthermore, we started to develop a multi-classes incremental svm classification, to be able to track several faces at the same time.

6 CONCLUSIONS

In this paper, we presented an on-line algorithm that makes use of human skin colour and texture properties, and uses an incremental svm classification for face tracking, under large lighting condition variations. The results obtained are very encouraging, and ameliorations are currently carried out. Our method could be applied on several applications, such videophone applications and video surveillance systems.

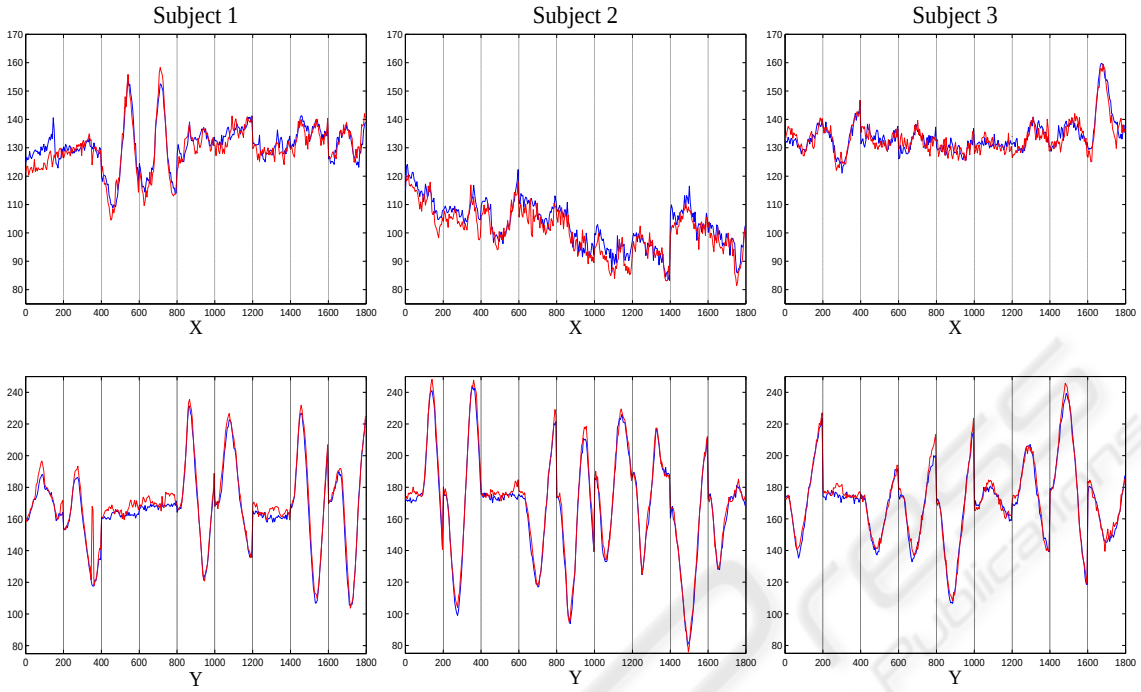

Figure 7: The curves of the evolution of the coordinates X and Y of the center of the ground truth faces (in red), superposed to those of the detected faces (in blue).

REFERENCES

Cauwenberghs, G. and Poggio, T. (2000). Incremental and decremental support vector machine learning. In *Neural Information Processing Systems*.

Chai, D. and Ngan, K. (1998). Locating facial region of a head-and-shoulders colour image. In *3rd IEEE International Conference on Automatic Face and Gesture Recognition*.

Chow, T., Lam, K., and Wong, K. (2006). Efficient colour face detection algorithm under different lighting conditions. *Journal of Electronic Imaging*, 15(1):013015.

Cula, O., Dana, K., Murphy, F., and Rao, B. (2005). Skin texture modeling. *International Journal of Computer Vision*, 62 (1-2):97–119.

Forsyth, D. and Fleck, M. (1999). Automatic detection of human nudes. *International Journal of Computer Vision*, 32(1):63–77.

La Cascia, M., Sclaroff, M., and Athitso, S. (2000). Fast reliable head tracking under varying illumination: An ap-proach based on robust registration of texture mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):332–336.

Martinkauppi, B. (2002). *Face Colour Under Varying Illumination -Analysis And Applications*. PhD thesis, Department of Electrical and Information Engineering and Infotech Oulu, University of Oulu.

McKenna, S., Raja, Y., and Gong, S. (1999). Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3-4):225–231.

Sigal, L., Sclaroff, S., and Athitsos, V. (2004). Skin colour-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):862–877.

Soriano, M., Martinkauppi, B., Huovinen, S., and Laaksonen, M. (2000). Using the skin locus to cope with changing illumination conditions in colour-based face tracking. In *IEEE Nordic Signal Processing Symposium*.

Strring, M., Andersen, H. J., and Granum, E. (1999). Skin colour detection under changing lighting conditions. In *7th International Symposium on Intelligent Robotic Systems*.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*.

Yang, M., Kriegman, D., and Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58.