FAST MULTI-VIEW FACE TRACKING WITH POSE ESTIMATION

Julien Meynet, Taner Arsan**, Javier Cruz Mota*, Jean-Philippe Thiran**

*Ecole Polytechnique Fédérale de Lausanne (EPFL),
Signal Processing Laboratories (LTS5),
1015 Lausanne, Switzerland,
{julien.meynet,javier.cruz,jp.thiran}@epfl.ch

**Kadir Has University,
Computer Engineering Department,
Istanbul 34230, Turkey,
arsan@khas.edu.tr

ABSTRACT

In this paper, a fast and an effective multi-view face tracking algorithm with head pose estimation is introduced. For modeling the face pose we employ a tree of boosted classifiers built using either Haar-like filters or Gauss filters. A first classifier extracts faces of any pose from the background. Then more specific classifiers discriminate between different poses. The tree of classifiers is trained by hierarchically sub-sampling the pose space. Finally, Condensation algorithm is used for tracking the faces. Experiments show large improvements in terms of detection rate and processing speed compared to state-of-the-art algorithms.

1. INTRODUCTION

Advances in the computing power of computers as well as the advances in image processing and pattern recognition made Computer Vision become a reality. In this point, it is very important to express that detection of human faces is a key component in applications of human-computer interaction. Briefly, the task of face detection is to find the position and size of one or more faces in an image or a video sequence.

Face detection techniques can be divided into two major classes: feature-based approaches and image-based approaches [1, 2, 3]. Feature-based approaches make use of face appearance properties, such as skin color or face geometry to detect faces. In recent studies, skin color based face detections have gained strong popularity. Color allows fast processing and is highly robust to geometric variations of the face pattern [4]. Low-level analysis, feature analysis, pixel-based skin color detection technique and active shape models are the examples of feature-based approaches. Image-based approaches use the advantage of the development and advances in the pattern recognition field of study, and treat the detection of faces problem as a general problem on pattern recognition. These approaches consider an image or a sub-image when they are searching for a face, as a whole object that has to be classified. Linear subspace methods or statistical methods are the examples of image-based approaches. Each of the two exposed families, feature-based methods and image-based methods has its own advantages and disadvantages. Feature based methods generally do not need complex model training and are robust to slight pose changes. However, in this work, we are interested in modeling the face pose which can only be achieved efficiently by image-based techniques. That is why, in the remaining of the paper, only the holistic methods will be considered.

The classical approach for holistic face detection is to scan the input image with a sliding window and for each position, the window is classified as either face or non face. The method can be applied at different scales (and possibly different orientations) for detecting faces of various sizes (and orientations). Finally, after the whole search space has been explored, an arbitration technique may be employed for merging multiple detections.

The first reference algorithm has been proposed by Sung and Poggio [5]. They use clusters of face and non face models to decide whether a constant sized window contains a face or not. The principle is to use several Gaussian clusters to model both classes. Then the decision is taken according to the relative distance of the sample to the mean of both classes. In order to detect faces at any scale and position they use a sliding window which scans a pyramid of images at different scales. The detector proposed by Schneiderman and Kanade [6] also models the probability distribution of the face class, but they employ a naive Bayes classifier. A similar holistic approach proposed by Rowley et. al. in [7] is one of the most representative for the class of neural network approaches. It comprises two modules: a classification module which hypothesizes the presence of a face and a module for arbitrating multiple detections. A fast algorithm is proposed by Viola and Jones in [8]. It is based on three main ideas. They first train a strong classifier by boosting the performance of simple rectangular Haar-like feature-based classifiers. They use the so-called integral image as image representation which allows to compute the base classifiers very efficiently. Finally [9] proposes to combine these Haar-like filters with more discriminant Gaussian filters and to apply a mixture of classifiers for improving the detection rates.

Most of these studies focused on frontal face detection, but similar techniques have been extended to multi-view face detection and face pose estimation. Pose estimation is usually achieved in two successive steps: first the face is located then its pose is estimated. Current state-of-the-art face detectors are very computationally efficient (e.g. [8]) and can lead to real-time applications by performing frame by frame detections. However it is preferable to perform specific tracking of the faces in order to increase the stability of the detections and to reduce the number of false detections.

According to [10], the face tracking methods are classified into three main groups: low level feature approaches, template matching approaches and statistical inference approaches. The low level feature approaches make use of low level face knowledge, such as skin color, or background knowledge such as background subtraction, rectangular features and motion information to track the faces. The template matching approaches involved tracking contours with snakes, 3D face model matching, shape and face template matching and wavelet networks matching. The statistical inference approaches include Kalman filtering techniques for unimodal Gaussian representations, Monte Carlo approaches for non-Gaussian nonlinear target tracking and Bayesian Network inference approaches. Recent researches have expressed that Kalman filtering techniques is in-adequate because it is based on Gaussian density. The Condensation Algorithm (conditional density propagation for visual tracking) uses factored sampling, previously applied to the interpretation of static images, in which the probability distribution of possible interpretations is represented by a randomly generated set. This algorithm uses learned dynamical models together with visual observations, to propagate the random set over time. The result is highly robust tracking of agile motions.

The purpose of this paper is two-fold. We propose a new tree-based structure of classifiers for performing jointly multi-view face detection and head pose estimation and we present an algorithm for tracking in real time both pose and position.

For modeling the face class, we use AdaBoost for combining base classifiers trained from Haar filters and Anisotropic Gaussian filters. Then hierarchical sub-sampling of pose space is considered and tree of classifiers are used for face pose estimation. Finally, Condensation Algorithm has been used for fast and robust

face tracking with pose estimation. This paper is organized as follows: Section 2 describes the multi-view face detection including face class modeling and pose estimation. Multi-view face modeling is explained in Section 3. Fast tracking with pose estimation algorithm is given in Section 4. Finally, experimental results are presented in Section 5 and Section 6 concludes the paper.

2. FACE CLASS MODELING

In this Section, we give a short overview of the Adaptive Boosting (AdaBoost) algorithm and we show how it can be used for performing feature selection too. Then we introduce the Haar-like filters and anisotropic Gaussian filters used for face modeling.

2.1 AdaBoost

AdaBoost [11] is a learning algorithm which iteratively builds a linear combination of some basic functions (*weak classifiers*) to form a strong ensemble:

$$f_T(\mathbf{x}) = \text{sign}\left(\beta_0 + \sum_{k=1}^{T} \beta_k h_k(\mathbf{x})\right), \tag{1}$$

with $h_k : \mathbb{R}^n \to \{\pm 1\}$ being the weak classifiers. Training in the case of AdaBoost comes to finding the weak classifiers and their corresponding weights β_k. For a detailed description of the algorithm the reader is referred to [11].

Finally, note that depending on the application we might prefer to favor one of the classes. In AdaBoost, this can be easily implemented by building an asymmetric version of AdaBoost that encourages the correct classification of the desired examples, and by tuning the final threshold β_0 on an independent set in order to obtain the desired operating point on the ROC curve.

Now let $\mathbf{x} \in \mathbb{R}^n$ be a vector whose components will be denoted by $x_j, j = 1, \ldots, n$. If we let the weak classifiers be

$$h_j(\mathbf{x}) = \begin{cases} 1, & \text{if } p_j x_j < p_j \theta_j \\ -1, & \text{otherwise} \end{cases}, \tag{2}$$

it turns out that AdaBoost will perform a feature selection too. Indeed, the final decision function will be a linear combination depending only on some of the features. This is the particular form of the weak classifiers that will be used for building the face detector and $\mathbf{x}$ will be the vector of all filter responses when applied to one image. For these weak learners (decision stumps), there are two parameters to be tuned: the threshold θ_j (chosen by maximum a posteriori rule) and the parity p_j.

2.2 Haar-like filters and Anisotropic Gaussian filters

In this section, we present the 2 types of filters used in this work for constructing the weak classifiers.

The Haar filters (HF) are made of 2, 3 or 4 rectangular masks with 2 scaling parameters and two center coordinates. The templates are shown in Figure 1(a). They were first introduced for frontal face detection by Viola et al. [8]. These filters can be applied very efficiently using a image representation called integral image.

However, they are not very discriminant for challenging examples and their rectangular shaped is not appropriated for pose variations. That is why we also use Gaussian filters (GF) that are more discriminant. They were first introduced by Peotta et al. in [12] for image compression and signal approximation. Then they were successfully employed for frontal face detection purposes in [13]. These filters are made of a combination of a Gaussian in one direction and its first derivative in the orthogonal direction. The generating function $\phi : \mathbb{R}^2 \to \mathbb{R}$ is given by:

$$\phi(x,y) = x \exp(-|x| - y^2). \tag{3}$$

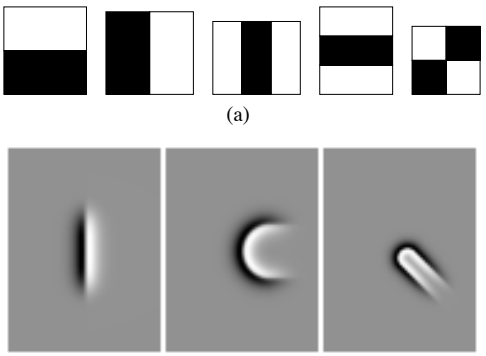

Figure 1: Filters used for modeling the faces: 1(a)Haar–like filters and 1(b)Anisotropic Gaussian filters.

It efficiently captures contour singularities with a smooth low resolution function in the direction of the contour and it approximates the edge transition in the orthogonal direction with the first derivative of the Gaussian.

In order to generate a collection of local filters, the following transformations can be applied to the generating function: translation($\mathscr{T}_{x_0,y_0}$), rotation($\mathscr{R}_\theta$), bending($\mathscr{B}_r$) and scaling in two directions($\mathscr{S}_{s_x,s_y}$).

By combining these four basic transformations, we obtain a large collection of functions $\mathscr{D} = \{\psi_{s_x,s_y,\theta,r,x_0,y_0}(x,y)\} = \{\mathscr{T}_{x_0,y_0}\mathscr{R}_\theta\mathscr{B}_r\mathscr{S}_{s_x,s_y}\phi(x,y)\}$. Figure 1(b) shows some of these functions with various bending and rotating parameters. We define the example $\mathbf{x}_k = (x_{jk})$ as the local responses of an image I_k to the all of the filters from $\mathscr{D}$:

$$x_{jk} = \iint \psi_j(x,y) I_k(x,y)\, dxdy \quad \forall \psi_j \in \mathscr{D}, \tag{4}$$

where the integral is taken over a suitable domain.

2.3 Gaussian vs. Haar–like features

Firstly, we are interested in comparing the Gaussian features (GF) [13] (see Figure 1(b) for an examples of such features) with the more commonly used Haar–like features (HF), introduced in [8] (Figure 1(a)). As we want to gain some insights about the intrinsic discrimination power of the two type of features, we trained two detectors using either the Gaussian filters or the Haar–like features, using the same training sets, and then we compared the two on an independent validation set. Figure 2 shows the classification performance of the two classifiers in terms of error rates. It is interesting

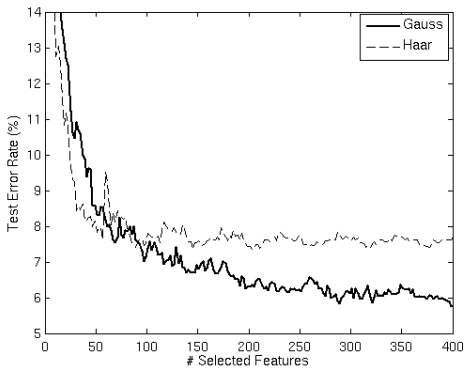

Figure 2: Performance of GF and HF–based detectors

Figure 3: Some of the first selected base functions.

to note from Figure 2 that while for the first ~ 100 iterations the error rate decreases quickly as we add more features to the model of the both classifiers, it remains practically constant for the HF–based detector. However, the GF model keeps improving, as we add more and more features. This shows that the HFs are not discriminant enough for modeling the finer differences between the two classes. Figure 3 shows the GFs that were selected during the first iterations so those that were deemed the most discriminative. Note also how they adapt to model the most salient features of the faces.

2.4 Cascaded classifiers

We have already noticed that the first features are good enough to produce an acceptable classifier. This observation led to the introduction of a multi–stage decision structure in which at each stage the number of non–faces that are rejected is maximized, constrained by having a low false negative rate. The system has a cascade (or serial) structure, with all the candidates that are not rejected by the i-th being fed into the $i+1$ stage. Finally, the candidates that are not rejected by the last stage are labeled as "faces". In our system, each stage was a classifier produced by boosting either the Gaussian or the Haar features.

Such design considerably improved the performances, as the latter stages can be tuned to correctly classify more complex candidates, without being distracted by simpler cases. In addition, the system becomes much faster, because the majority of the candidates is rejected using a few operations.

3. MULTI-VIEW FACE MODELING

In previous Section, two kinds of filters have been shown. On the one hand, Haar- based filters have very low computing requirements but are not selective enough to distinguish between faces with different poses. On the other hand, Anisotropic Gaussian filters are selective enough to distinguish between faces with different poses but the computation of its response is more costly. In order to take advantage of both filter types, we propose to use a tree of classifiers that benefit from the efficiency of HF and discriminant power of GF.

3.1 General Classifier

The first classifier in the tree finds quickly potential face candidate. Thus, it was trained to classify between faces of any pose and non face images. As in classical applications a large majority of scanned windows are non-face examples, this task needs to be performed very efficiently. That is why we use Haar filters for building this classifier. In the following it will be referred as General classifier (GC). In the different tests that have been done with the system described here, the GC discards around 95% of the windows tested.

3.2 Pose Classifiers

Then the remaining face candidates are presented to the next classifiers in the tree. The aim of these classifiers is to reject false positive windows but also to discriminate between different poses. That is why they are trained using the more discriminant Gaussian filters. For this we use a hierarchical classification scheme. Let us first define the pose space as the space spanned by two angles: elevation $\phi \in [-\frac{\pi}{3} \frac{\pi}{3}]$ and out-of-plane rotation $\theta \in [\frac{\pi}{2}, \frac{\pi}{2}]$. This pose space covers from left profile ($\theta = -\frac{\pi}{2}$) to right profile ($\theta = \frac{\pi}{2}$). Moreover, an elevation of $\frac{\pi}{3}$ is sufficient to model most of the face movements. Each single point in this space represents a specific pose (the origin (0,0) corresponds to frontal pose). This space is sub-sampled iteratively in smaller regions. We introduce slight overlaps between neighbor regions to avoid boundary problems. This will allow to better estimate the pose by combining outputs from different classifiers. Then, for each region we train a so-called pose classifier(PC) with face images only corresponding to the specific poses of the region-of-interest, against non face images. This hierarchical process for building the tree is represented in Figure 4. Each classifier PC is thus a cascade of GF that will reject all windows being either non-faces or faces of different poses. We obtain a tree of classifiers with more and more pose specific classifiers as descending towards the leafs. The global structure of the tree is represented in figure 5. The advantages of this structure are two-folds. It first efficiently discards non-face windows. In fact, each PC is trained to discard the specific non-face windows that have been incorrectly classified by the previous layer of PC. The tree acts as a global cascade of classifiers that significantly reduces the number of false positives. On the other hand the output of the classifiers of the last layer will give an estimate of the head pose, thus producing a simultaneous face detector and head pose estimator.

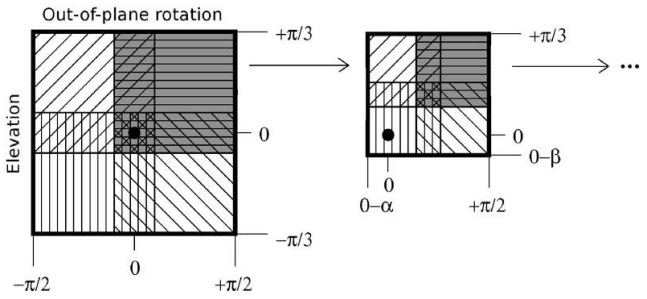

Figure 4: Hierarchical sampling of the pose plane.

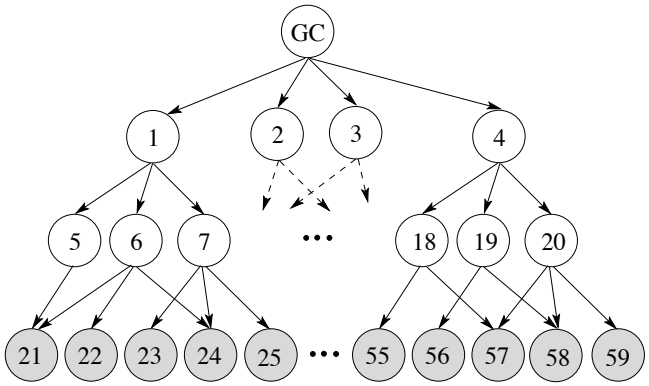

Figure 5: Tree structure for multi-view pose detection. GC is built using HF. All the other pose specific classifiers use GF.

3.3 Detection Process

A candidate window presented to a PC will be rejected if it is classified as non face or if it is classified as face with a wrong pose. Otherwise it is given to the children classifiers in the tree. At the end, if several leafs classify the candidate as face, only the pose with maximum posterior probability is kept.

This multi-view face detector is efficient and can be used in video applications by performing a complete detection in each frame. However, next section will introduce a tracking system that improves the pose estimation performances while increasing the detection speed so that the system can be applied in real-time applications.

4. FAST TRACKING WITH POSE ESTIMATION

For tracking the multi-view faces, we used an efficient probabilistic tracking framework. The algorithm is based on the Condensation (conditional density propagation for visual tracking) [14].

4.1 Condensation

The condensation algorithm is an extension of the factor sampling to a sequence of images (see [14] for details). It approximates probability distributions of likely object states. This allows the tracking of multiple instances contrary to a Kalman filter. We denote by $\mathbf{x}_t$ the object state at time t and by $\mathbf{z}_t$ the observation at time t. The principle is to approximate the conditional state density $p(\mathbf{x}_t|\mathbf{z}_t)$ by a weighted sample set $\{(s_t^{(n)}, \pi_t^{(n)})\}$. In a first frame, we generate N hypothesis $s_0^{(n)}$ and assign to each of them a weight defined as $\pi^{(n)} = p(Z|X = s_0^{(n)})$, according to a known prior $p(X)$. Then the probability densities are propagated in 3 stages: selection, prediction and measurement.

At time-step $t+1$, a new sample set $s_{t+1}^{(n)}$, is generated from $s_t^{(n)}$ according to the weight distribution $\pi_t^{(n)}$. The samples the most likely to represent the object (with highest probability $\pi_t^{(n)}$) may be reproduced several times while samples with low probability may not be chosen at all. At the prediction stage, this new sample set is diffused according to a predefined dynamical model. Finally new probabilities $\pi_{t+1}^{(n)}$ are estimated from the measurements $\mathbf{z}_{t+1}$.

4.2 Face Position and Pose Tracking

In order to adapt the Condensation algorithm to our face tracking system, we need to design two models: the prediction model for diffusing the sample set and the measurement metrics for estimating new sample weights. First let us define the object state in our application. A face state is represented by its position $\mathbf{u} \in \mathbb{R}^2$ and its pose $\psi \in [-\frac{\pi}{2}\frac{\pi}{2}] \times [-\frac{\pi}{3}\frac{\pi}{3}]$. In order to speed up the tracking, we tackle the problem in two successive steps: we first track the position of the face, then we track the pose at the found position. The consequence of this is that we need much less samples to approximate efficiently the distributions. As we do not have any prior constraint on the face dynamics during the sequence, we use a simple Gaussian diffusion at each time-step. For tracking the position $\mathbf{u}$, we only apply the 5 first classifiers in the tree (1 GC + 4 PCs), as the other pose specific PCs are not relevant for modeling the position of the face. Then for tracking the pose ψ we apply classifiers in a single branch of the tree depicted in Figure 5 (1 classifier in each layer from GC to 1 leaf).

Finally the weight $\pi_t^{(i)}$ of each sample $s_t^{(i)}, i = 1, \ldots, N$, is estimated from the output confidence of the last classifier applied in the tree.

This tracking strategy reduces significantly the number of classifiers to be computed in each frame. In fact we first restrict the search to windows around the initial positions of the faces and then for each window we do not need to apply the complete tree for estimating the pose. Thus, tracking faces reduces significantly the computational cost.

5. EXPERIMENTS

In this Section, we report a number of experiments that we performed for assessing the performance of the proposed tracking system. The detectors we built were following a classical scheme, in which the image was explored at different positions and scales using a sliding window. At each position (in image and scale space) the window is classified as either face or non–face. At the end, multiple detections of the same face must be arbitrated. We simply took the average detection even if it leads to a slightly less precise detection. Concerning the initialization of the tracker, a complete detection is performed in the first frame in order to find all the faces. We also reinitialize the system by a complete detection every 2 seconds in order to detect potential new faces entering into the scene.

5.1 Data used and model training

As it was pointed out in [15], a good window size for the scanning window is 20×20 pixels. That is why in this work, all training samples and candidate windows are re-scaled to this base size, in order to detect faces at any size. For training the tree of classifiers, we used images from two face datasets: the CMU Pose, Illumination , and Expression (PIE) database [16] and INRIALP [17]. They contain numerous face images at different poses and under various illumination conditions. We used $47,954$ images from PIE and $2,597$ images from INRIALPES. Finally we used FERET dataset [18] as a validation set for adjusting the following hyperparameters: the thresholds of each classifiers, the number of classifiers in each cascade and for estimating the true positive rates and false positive rates after each PC.

The set of negative examples (non–faces) was built by bootstrapping from randomly selected images which contained no faces and it contained about $500,000$ examples.

The systems contains a GC made of 4 stages of Haar features (a total of roughly 150 HF) and a total of 59 PC. Each PC is made of 4 stages of GF with, in average, 75 features in each PC.

The tracking system has been tested on various sequences containing one or several moving faces. These video sequences are available upon request.

5.2 Accuracy of the system

The complete tracking system has been tested on several sequences, with various background and illumination conditions. Some resulting frames are given in Figure 6. It turns out that the tracking of the face position is very precise even with very complex backgrounds. The tracking is also much more stable than a frame-by frame detection. Concerning the tracking of the pose we obtain 93% of correct classification. A estimation is considered correct if the difference between the detected angle and the true angle is at most 10 degrees (and supposing that the position of the face is tracked correctly). We obtain very good results for the out-of-plane rotation, while the elevation is slightly less accurate. This can be explained by the defects of the datasets used for training the system. In fact, the elevation angles are not always labelled identically from one dataset to another. In some datasets, the annotation was done by asking persons to look at specific markers put at precise angles. The risk is that the person may move the eyes for looking at the marker which introduces a noise in the dataset. In other datasets, the person's head is fixed and the cameras are moving to specific angles. But the resulting images do not correspond to real out-of-plane rotation or elevation seen from the front of the person.

5.3 Time performances

Performing a tracking of the faces improves the performances in terms of accuracy but also in terms of processing speed. In fact, tracking faces from frame to frame avoids to perform an exhaustive search in the image. Moreover, for each candidate window, a maximum of 5 classifiers are applied for tracking the position and 4 for the pose, instead of the 60 classifiers present in the complete tree.

We give hereafter a speed comparison for two detectors. The first system performs a complete multi-view detection in each frame of the video while the second system tracks the faces as described in the previous section. We apply these 2 detectors on a sequence of $1,500$ images with 320×240 pixels, each frame containing one or several faces. We then report in Table 1 the average detection speed (number of frames per seconds). This shows that the tracking system is much more computationally efficient than a detection frame by frame.

5.4 Discussion

Researchers, who are interested in face detection and tracking algorithms, use their own patterns and videos instead of using a publicly

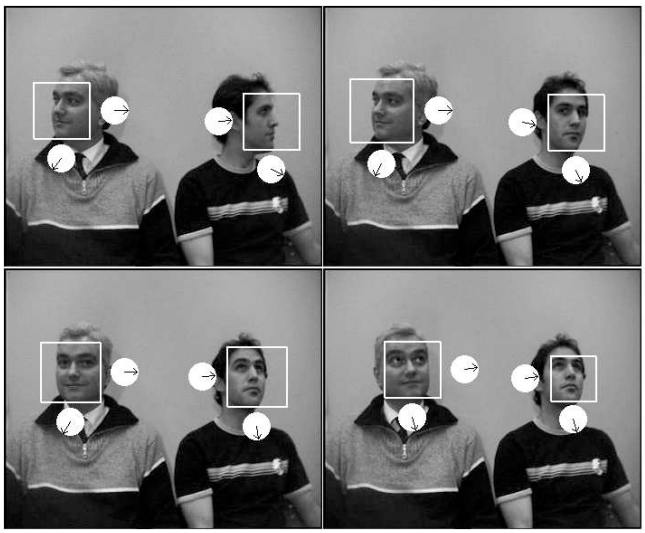

Figure 6: Some examples of detections. The 2 circles nearby each window represent the pose. The upper circle is the elevation, the other represents the out-of-plane rotation.

Table 1: Detection speed in frames per seconds (fps) of 2 detectors. The measure is an average over the 1,500 frames of a sequence of 320×240 pixels images.

Detector	fps
Face detection frame by frame	6.36
Face tracking	23.45

available common content, while they are expressing experimental results. We also use our own video sequences and give the results on these videos. Therefore, it is not possible to make an exact comparison between our algorithm and other proposed face tracking algorithms.

6. CONCLUSIONS

In this paper, we propose a system for fast multi-view face tracking in video sequences. The modeling of the face at different pose has been performed using a tree of classifiers. The first group of classifiers are computationally very efficient and they discard the background regions quickly. Then, more specific classifiers have been trained with Gaussian filters to distinguish between different poses. The output is thus the position and pose of the detected faces. A tracking algorithm based on Condensation has been used for tracking the faces in time. Some experiments show the accuracy of the system on real world sequences.

REFERENCES

[1] M. Yang and D. J. Kriegman and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp.34–58, 2002.

[2] E. Hjelmas, B. K. Low, "Face Detection: A Survey," *Computer Vision and Image Understanding*, vol. 83, no.39 pp.236–274, September 2001.

[3] T. Sauquet, Y. Rodriguez and S. Marcel, "Multiview Face Detection," *IDIAP*, IDIAP-RR, no:49, 2005.

[4] V. Vezhnevets, V. Sazonov and A. Andreeva, "A survey on pixel-based skin color detection techniques," *Proc. Graphicon-2003*, 2003.

[5] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[6] H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Los Alamitos, 2000, pp. 746–751.

[7] H. A. Rowley, S. Baluja, and T. Kanade, "Human face detection in visual scenes," in *Advances in Neural Information Processing Systems*, David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, Eds. 1996, vol. 8, pp. 875–881, The MIT Press.

[8] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[9] J. Meynet and V. Popovici and JP. Thiran, "Mixtures of Boosted Classifiers for Frontal Face Detection," *Signal Image and Video Processing*, 2007, In Press.

[10] H. Wu and J. S. Zelek, "The Extension of Statistical Face Detection to Face Tracking," 1st Canadian Conference on Computer and Robot Vision (CRV'04), pp.10–17, 2004.

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[12] L. Peotta, L. Granai, and P. Vandergheynst, "Very low bit rate image coding using redundant dictionaries," in *Proc. of the SPIE, Wavelets: Applications in Signal and Image Processing X*, November 2003, vol. 5207, pp. 228–239.

[13] J. Meynet and V. Popovici and JP. Thiran, "Face detection with boosted gaussian features," *Pattern Recognition*,Vol. 40, Nr. 8, pp. 2283-2291, 2007.

[14] A. Blake and M. Isard "The CONDENSATION Algorithm - Conditional Density Propagation and Applications to Visual Tracking," in *MIT Press, NIPS*, 1996, pp. 361–367.

[15] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proc. IEEE International Conf. on Image Processing*, 2002, pp. 900–903.

[16] T. Sim and S. Baker and M. Bsat, "The CMU Pose, Illumination, and Expression Database," in *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.25, number 12, 2003, pp 1615-1618.

[17] N. Gourier and D. Hall and J. L. Crowley, " Estimating face orientation from robust detection of salient facial features," in *Proc of Pointing 2004, International Workshop on Visual Observation of Deictic Gestures*, 2004.

[18] P.J. Phillips and al., "The FERET database and evaluation procedure for face-recognition algorithms," in *Image and Vision Computing*, 1998, vol. 16.