Towards A Robust and Real-time Face Detection and Tracking Framework

Peihua Li
IRISA/INRIA Rennes, France, Email:pli@irisa.fr
Computer Department, Heilongjiang University, China

Kai Xie
Harbin Normal University, China
Computer Department

Abstract

Traditional research and development in face detection and tracking are mostly focused on either face detection in still images, or face tracking in image streams, without remarkable combination of both. As a result, few impressive face detection and tracking systems can be seen nowadays. The paper contributes to effectively incorporation of two modules, detection module and tracking module, into one integrated and coupled framework. In the detection module we construct AdaBoost detectors for face detection in video stream, which are able to detect multi-view faces and are robust to illumination changes in video stream. The tracking module is based on the improved mean shift tracking, which has advantage of target model adaption. The detection module, triggered every ten frames, is not only responsible for initiating a tracking thread, but also, together with tracking module, contributes to target model update. The system is able to benefit from the strengths of different techniques and overcome their respective limitations. Experiments on real image sequences show that the system is robust and real-time in realistic environment.

1. Introduction

The issues of face detection and tracking have long been studied in the fields of computer vision and pattern recognition. The widespread interests devoted to such research and development, are due in part to increasingly growing performance/price ratio of computing power and related hardware, and beyond that, due to potential important applications in surveillance, human-computer interaction, retrieval among others.

Many researchers have been attempting to constructing realistic face detection and tracking systems. Spors and Rabenstein present a real-time face localization and tracking algorithm for color video [1]. The face localization is based on skin color segmentation, and tracking is accomplished through Kalman filtering, which estimates the position and size of face with the help of eye localization based on PCA. The performance of face detection relying on skin color is, however, easily affected by lighting conditions. Comaniciu and Ramesh propose an efficient framework for detection and tracking of human faces [2]. Both detection and tracking are based on mean shift algorithm, which aims at optimization of a metric function between two distributions. Yet they assume that the target statistics is ad-hoc and unvaried, which is not the case in real-world applications. CAMSHIFT algorithm [3] intends to tracking face for use in a perceptual user interface. The algorithm depends on skin color projection onto flesh probability image and on gradient optimization method. Because of lack of a distance metric target distributions, it is not scale-invariant [6]. In addition, it has aslo assumption of prior-knowledge of face size and its histogram. Shakharovich et al describes a unified learning framework for face detection, tracking and classification [4], in which Viola-Jones AdaBoost detector is not only used to perceive faces, but also used as an observation probability of an image patch. Particle filtering was introduced to improve tracking performance. Nevertheless, tracking often fails because frontal face detector is unable to deal with face pose variations.

Unlike the above-mention work, we present a novel framework consisting of two independent but interactive modules—face detection module and tracking module. The detection module concerns a boosted multi-view face detector, triggered each ten frames, running in a background thread with lower priority, while tracking module is time critical and runs in a thread with higher priority, based on Model–Adaption Mean Shift algorithm. Furthermore, the two modules are interactive to one another, responsible for the update of target distributions.

2. Boosted Multi-view Face Detection in Video

Face detection approaches in static images has made considerable advance in recent years [8], such as those based on manifold analysis, Neural Networks, Support Vector Machines (SVM), or sparse networks of winnow. A more recent and impressive work is that of Viola and Jones [5], which achieves comparable frontal face detection re-

sults with other well-known face detectors while is approximately 15 times faster than any previous approach. Based on AdaBoost algorithm [9], their work realizes the selection of critical visual features from a large set of Harr-like features and the training of AdaBoost simultaneously.

One of the main ideas of the AdaBoost algorithm is to maintain a distribution or a set of weights over the training set, by calling a given weak learning algorithm repeatedly in a series of rounds $t = 1, \cdots, T$. The weights on training examples on the round k is denoted $D_k(i)$ $i = 1, \cdots, N$. Denote training examples $\{(\mathbf{x}_i, y_i), i = 1, \cdots, N\}$, where $\mathbf{x}_i$ is an image patch of fixed size, and $y_i = 1$ for face examples and $y_i = -1$ for non-face examples. Initially, weights are set equally among face and non-face examples. But on each round, the weights of incorrectly classified examples are increased, so that in the later consecutive training stages the weak learner is forced to focus on the hard examples in the training set. More precisely, the weak learning algorithm's job is, among a set of weak functions $f_j(\mathbf{x}_i), j = 1, \cdots, M$, to select one feature $h_k(\mathbf{x}_i) = f_{j^*}(\mathbf{x})$ which satisfies the following equation

$$p_{D_k}(f_{j^*}(\mathbf{x}_i) \neq y_i) = \mathrm{argmin_j} p_{D_k}(f_j(\mathbf{x}_i) \neq y_i) \quad (1)$$

where $p_{D_k}(f_j(\mathbf{x}_i) \neq y_i) = \sum_i^N D_k(i)\delta(f_j(\mathbf{x}_i) \neq y_i)$. Once the weak learner h_k has been received, AdaBoost chooses α_k as follows

$$\alpha_k = 0.5 \ln(1/q_k - 1) \quad (2)$$

where $q_k = p_{D_k}(h_k(\mathbf{x}_i) \neq y_i)$, which measures the importance that it assigns to h_k. The distribution D_k is then updated using the rule as below

$$D_{k+1}(i) = D_k(i) \exp(-\alpha_k y_i h_k(\mathbf{x}_i))/Z_k \quad (3)$$

where Z_k is a normalization factor. The final hypothesis H is an average of the T weak hypotheses

$$H(\mathbf{x}) = \sum_{k=1}^{T} \alpha_k h_k(\mathbf{x}) \quad (4)$$

2.1. Multi-view Face Detector in Video

Our multi-view face detector consists in two level pyramid and six cascades, responsible for five different views: frontal view, left-half and left profiles, right-half and right profiles. The first level concerns a cascaded face detector trained on all training examples, which contains non-face examples and face examples in five different views. The test image regions which only pass the first level will continue to try to pass the second level. Five different cascaded detectors are in the second level which are responsible for detections of faces which may be in different views. There are a total of six cascaded classifiers need to be trained. Details about the algorithm may be referred to [7].

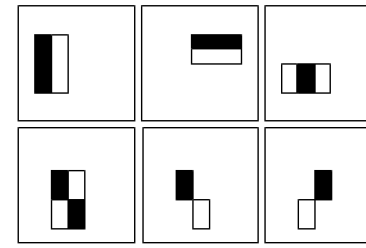


Figure 1. Six type of features relative to the surrounding detection windows. A feature is defined as the difference between the pixels sum lying within the white rectangles and that in the black rectangles.

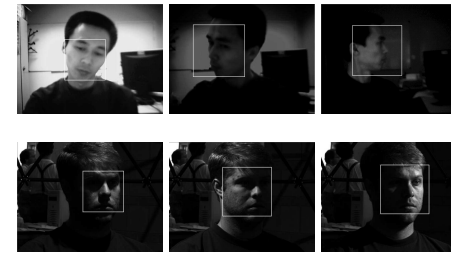


Figure 2. Some of detection results with our multi-view face detector

For the purpose of dealing with non-symmetric features in multi-view faces, we use six rectangle features as shown in Figure 1. To make the face detector not sensitive to illumination changes, we include both face and non-face training examples that accommodate significant illumination variations. It should be noted that our purpose is to detect human faces in video instead of static images. We therefore train the multi-view detector such that the false alarm rate is low (and, of course, the detection rate is low also), for we do not want to see some false objects being detected and then tracked. Although one face may be missed in the current frame due to low detection rate, there will have a good possibility in the subsequent frames for the face to be perceived , which is quite different from face detection in static frames.

Figure 2 shows some of detection results with our multi-view face detector. The upper row images show that multi-view faces are detected with a degree of in-plane rotation and upside-down rotation. The lower row images, where, from the left to the right, the light casts from the upper-front, the left and the right of the subject respectively (Figures courteous of [10]), show that our face detector is robust to illumination variations.

3. Model–Adaption Mean Shift Face Tracking

The core of mean shift tracking consists in the mean shift iteration [6], based on the similarity of target distribution $\mathbf{q}$ and candidate distribution $\mathbf{p}$, represented respectively by multi-channel color histograms $\mathbf{q} = \{q_u\}$ and $\mathbf{p} = \{p_u\}$, where $u = 1, \ldots, m$ indicate the histogram bins. Let us denote $\mathbf{z}_i$ $i = 1, \ldots, n$ the pixel locations of one face candidate, centered at $\mathbf{y}$ in the current frame, the distribution of the face candidate can be expressed as

$$p_u(\mathbf{y}) = \frac{1}{\sum_{i=1}^{n} k(\|\frac{\mathbf{y}-\mathbf{z}_i}{\mathbf{h}}\|^2)} \sum_{i=1}^{n} k(\|\frac{\mathbf{y}-\mathbf{z}_i}{\mathbf{h}}\|^2)\delta(b(\mathbf{z}_i)-u) \tag{5}$$

where $\mathbf{h}$ is the radius of a candidate region, $b(\mathbf{z}_i)$ is a function which associates to the pixel at location $\mathbf{z}_i$ the index $b(\mathbf{z}_i)$ of the histogram, and $\delta(\cdot)$ is the Kronecker delta function. The weighting function is adopted as Epanechnikov kernel.

The similarity between target and candidate distributions is measured by a scale-invariant metric function defined as follows

$$d(\mathbf{q}, \mathbf{p}(\mathbf{y})) = \sqrt{1 - \rho(\mathbf{q}, \mathbf{p}(\mathbf{y}))} \tag{6}$$

where $\rho(\mathbf{q}, \mathbf{p}(\mathbf{y})) = \Sigma_{u=1}^{m} \sqrt{p_u(\mathbf{y}) q_u}$ is Bhattacharyya coefficient. The mean shift iteration consists in the Steepest Descent minimization of Equation (6), derivation of which is omitted here.

3.1. Target Model Adaption

Because target distribution is expressed as a multi-channel color histogram, mean shift algorithm is sensitive to illumination changes [11]. It is thus of great importance to adapt target model. Here the difficulty is that how we decide when we should make such an adaption. Nummiro et al. propose that update should be made if the target model is significantly different from the tracking result. But if the tracking result is not accurate, for example, tracking is distracted either because of illumination variations or attraction by nearby image patches with similar color with target (but target is not lost), this kind of update criteria will deteriorate rapidly. We depend on the detection module for target model adaption, which will well cope with the above problem, since it is, as mentioned earlier, quite robust to illumination changes.

Let $\mathbf{q}'$ be the distribution of the target model, $\mathbf{p}$ the distribution of the same object outputted by the detection model, if below equation

$$d(\mathbf{q}', \mathbf{p}) < d_{thre} \tag{7}$$

holds, where d_{thre} is a distance threshold, we adapt, under the assumption that the target distribution $\mathbf{q}$ varies smoothly, the target model according to the following equation

$$\mathbf{q} = (1 - \alpha)\mathbf{q}' + \alpha\mathbf{p} \tag{8}$$

where α is a forgetting scalar. We set $\alpha = 0.85$ in the paper emphasizing the importance of newly detection result.

4. Face Detection and Tracking in One Framework

The framework consists of two modules: detection module and tracking module, as shown in Figure 3, both running independently while being interactive to one another.

4.1. Detection Module

The detection module is triggered every ten frames. The boosted multi-view face detector performs an exhaustive search in the image—scan the whole image at a step size of 1.2 pixels and all possible scales ranging from the minimum size of 20×20 pixels to the size of the whole image. The exhaustive search generally takes a long time, varying according to the number of the human faces present because of the cascade structure. The average time is about 200ms when up to three faces present. As the detection module is a long time-consuming procedure, it is arranged in the program to run in a background thread with lower priority. The output of detection module is a group of squares $\mathbf{r}_d = (x_d, y_d)$ with width and height $\mathbf{h}_d = (h_d, h_d)$, each of which corresponding to a detected human face.

4.2. Tracking Module

The tracking module maintains a list of objects. For each object, when a new frame is available we perform the following tracking algorithm.

1. Given the target distribution $\mathbf{q} = \{q_u\}_{u=1,\ldots,m}$ and its position $\mathbf{y}_{k-1}$ and scale $\mathbf{h}_{k-1}$ in the previous frame $k-1$.
2. For $i = 1$ to 3
   a. Set $\mathbf{h}_k^{(i)} = \mathbf{h}_k + 0.2 * (i-2) * \mathbf{h}_k$
   b. Let $\mathbf{y}_k^{(i)} = \mathbf{y}_{k-1}$, calculate $\mathbf{p}(\mathbf{y}_k^{(i)})$ and the new location as follows

$$\mathbf{y}'_k = \frac{\sum_{i=1}^{n} \mathbf{z}_i \sum_{u=1}^{m} \delta(b(\mathbf{z}_i) - u)\sqrt{q_u / p_u(\mathbf{y}_k^{(i)})}}{\sum_{i=1}^{n} \sum_{u=1}^{m} \delta(b(\mathbf{z}_i) - u)\sqrt{q_u / p_u(\mathbf{y}_k^{(i)})}}$$

   c. If $\|\mathbf{y}_k^{'} - \mathbf{y}_k\| > \varepsilon = 1$, set $\mathbf{y}_k^{(i)} = \mathbf{y}'_k$,then go to step b; otherwise, set $\mathbf{y}_k^{(i)} = \mathbf{y}'_k$, next i.

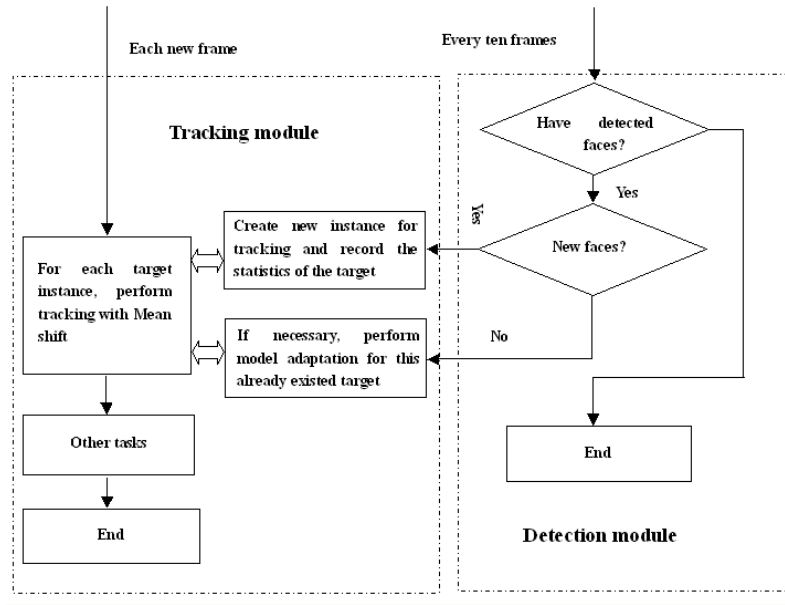


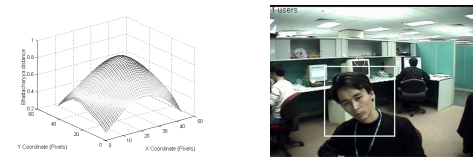
Figure 3. Demonstration of framework for face detection and tracking.

4. Get the last converged state of the object $\mathbf{y}_k = \mathbf{y}_k^{(i^*)}$ and $\mathbf{h}_k = \mathbf{h}_k^{(i^*)}$, where i^* satisfies the following equation

$$d(\mathbf{q}, \mathbf{p}(\mathbf{y}_k^{(i^*)})) = \min_{i=1,2,3} d(\mathbf{q}, \mathbf{p}(\mathbf{y}_k^{(i)}))$$

As the tracking undertaking is time-critical, we create in the program an independent thread running with higher priority for the tracking module.

The mean shift tracking algorithm is computationally efficient. Figure 4 presents the illustration of Bhattacharyya coefficients. The left figure shows Bhattacharyya coefficients corresponding to, in the right image, the white rectangle region containing a subject face. Note that the surface is very smooth and the peak identifies the face location. Because we adopt color histogram to capture the characteristic of face, the algorithm is free of facial expressions and pose variations, including profile to frontal rotation, upside-down rotation, in-plane rotation, or combination of them.



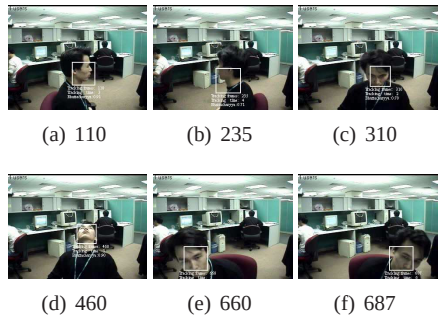
(a) (b)

Figure 4. Illustration of Bhattacharyya coefficients. The left figure shows Bhattacharyya coefficients corresponding to, in the right image, the white rectangle region containing a subject face. Note that the surface is very smooth and the peak identifies the face location.

4.3. The Framework for Face Detection and Tracking

Note that the output from detection module includes both new faces just entering the view or ones already perceived previously. The detection module, together with the tracking module, will make a judgement to see whether or not it is a new face. If yes, the tracking module will create an instance to trace this target, and at the same time, the statistics of the target is recorded; otherwise, it is decided whether target model adaption should be performed. The tracking module is in charge of tracking of each target and also knows about its state, for example, whether it is lost. In that case, the tracking instance for that object should be terminated.

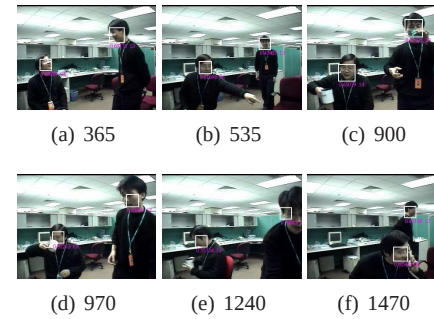
For each detected square region denoted by $\mathbf{r}_d$ and $\mathbf{h}_d$ indicating a face present, one target is initialized with the position $\mathbf{y}_0 = \mathbf{r}_d$ and scale $\mathbf{h}_0 = (h_x, h_y) = (h_d, 1.2h_d)$ and its distribution is calculated. By comparing the degree of overlapping between the rectangle $(\mathbf{y}_0, \mathbf{h}_0)$ and that of each of all existed face regions, we determine whether it is a new one. If the overlapping region with one of existed face rectangle is more than half of the target size, this detected rectangle represents a new face; otherwise, the distance between the distribution of the detected one and that of the already existed one is calculated, if Equation (7) satisfies, target distribution adaption should occur according to Equation (8).



(a) 110 (b) 235 (c) 310

(d) 460 (e) 660 (f) 687

Figure 5. Some of results in the first video stream with RGB 16 by 16 by 16 (Mean tracking time is 4ms)



(a) 365 (b) 535 (c) 900

(d) 970 (e) 1240 (f) 1470

Figure 6. Some of results in the second video stream with RGB 32 by 32 by 32 (Mean tracking time is 22ms)

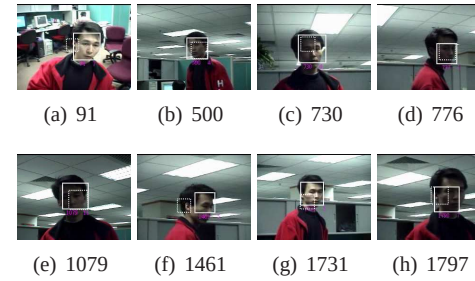
5. Experiments

To demonstrate the effectiveness of the proposed framework, we make experiments in three video streams (sampling rate is 25Hz), recorded in a typical office environment with pan, tilt and zoom (PTZ) camera which is placed on the top of the PC. In the first two video streams, the camera is motionless while the subject is moving, and in the third video stream, visual servoing is involved while both the camera and the subject are moving. The program is implemented with Visual C++ on a Pentium IV 1.2 GHz PC with Microsoft Windows 2000. Although some researchers are in favor of normalized RG space for target distribution, we do not see improvement with this. We instead use RGB space to make use of more information to characterize target density.

In the first video stream, sitting in front of the PC, the subject undergoes significant poses variations: changing from frontal view to profile or vice versa, or in poses with in-plane and upside-down rotation. The system can robustly detect and track the object. Some of typical results are presented in Figure 5.

The second experiment concerns two individuals freely talking, moving, drinking and calling. In the video stream, the pose variation and facial expressions of the subjects are significant, and also occurs partial occlusion when the subjects drink and make a call. Despite these, the system works well, as shown by typical tracking results in Figure 6.

The third experiment is concerned with face tracking based on visual servoing. The task involves controlling PTZ of camera, so that the target lies in the middle of the image as possible, and the size of which is of eligible size. Because of the motion of the camera and that of the subject, the illumination changes are considerable. We do not turn to advanced visual servoing technology, instead, we use very simple method: through a peripheral communication port, the host computer emits commands to command the camera to pan, tilt or zoom, according to the tracked result.



(a) 91 (b) 500 (c) 730 (d) 776

(e) 1079 (f) 1461 (g) 1731 (h) 1797

Figure 7. Some of results in the third video stream with RGB 16 by 16 by 16 (Mean tracking time is 6ms). Solid rectangles represent results with model-adaptation (MAD) mean shift, while the dotted ones with model-adaptation-free (MAF) mean shift.

With our detection module, we compare the performance of tracking algorithm between the proposed one–model-adaptation (MAD) mean shift with model-adaptation-free (MAF) mean shift [6]. As of MAF mean shift, in the whole image stream, the object is lost in frames 98, 277, 327 and 1402 respectively, because of considerable lighting changes. After the object is perceived by our detection module, the tracking is initiated and continued again with MAF mean shift. The target is, however, declared as a completely different one from the previous, and the identity of the target is therefore lost. In contrast, MAD mean shift is capable of tracking the target all over the image sequence, maintaining a stable identity of the object. In addition, even the MAF mean shift manages to track object in most cases, the locating of the target is inaccurate. Figure 7 shows some typical results with MAD algorithm, displayed with solid rectangles, and those with MAF algorithm, displayed with dotted rectangles.

To adapt to illumination variations, there is a total of 12 times of model adaption occurring in the image stream for the proposed tracking algorithm (MAD), namely, frames

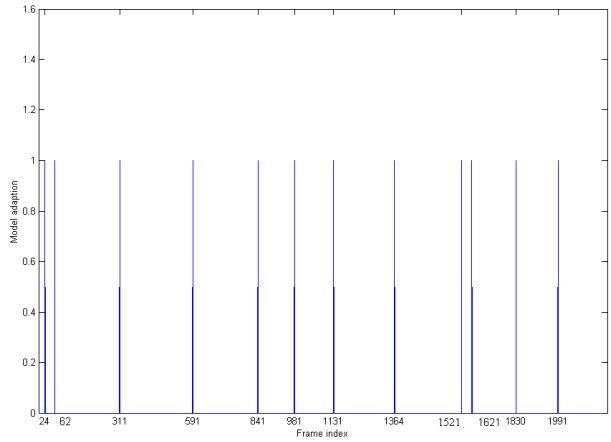


Figure 8. Illustration of model adaptations for the proposed tracking algorithm. Model adaption occurs in frames 24, 62, 311, 591, 841, 981, 1131, 1364, 1521, 1621, 1830 and 1991.

24, 62, 311, 591, 841, 981, 1131, 1364, 1521, 1621, 1830 and 1991, as illustrated by Figure 8. In the figure, the horizontal coordinate represents the frame index, and the value of vertical coordinate equal to 1 indicates that model adaption occurs in that frame.

6. Conclusions

In this paper, we propose a robust real-time face detection and tracking methodology that works well under realistic environment. The contributions of the papers are as follows:

- Effective combination of face detection module and tracking module into one integrated and interactive work.
- Construction of detector for face detection in video stream, which is able to detect multi-view faces with a degree of in-plane rotation and upside-down rotation, and is insensitive to lighting changes
- Presenting of an improved mean shift tracking algorithm—model-adaption (MAD) mean shift, which is robust to illumination variations and partial occlusion, as well as face pose and expression variations

Extensive experiments show that the framework is computationally efficient, insensitive to face pose and expression variations, immutable to partial occlusion, and robust to illumination changes.

Acknowledgments

Part of the work was carried out while the first author visited Microsoft Research Asia. Dr. S.Z. Li was acknowledged for providing this visiting opportunity. The first author would also like to thank Mr. Guofei Gu, whose collaboration and patience made it possible to record the second image sequence.

References

[1] S. Spors and R. Rabenstein. A Real-time Face Tracker of Color Video. In *IEEE Int. conf. on Acoustics, Speech and Signal processing*, Utah, USA, May 2001.

[2] D. Comaniciu, V. Ramesh. Robust Detection and Tracking of Human Faces with an Active Camera. In *IEEE Int. Workshop on Visual surveillance*, Dublin, Ireland, pages 11-18, 2000.

[3] G.R. Bradski. Computer Vision Face Tracking for Use a Perceptual User Interface. *Intel Technology Journal*, 2(2):12-21, 1998.

[4] G. Shakhnarorich, P.A. Viola and B. Moghaddam. A Unified Learning Framework for Real Time Face Detection and Classification. In *IEEE Int. Conf. on Automatic Face and Gesture Recoghition*, 2002.

[5] P. Viola and M.J. Jones. Robust Real-time Oject Detection. In *IEEE Workshop on Statistical and Computational Theories of Vision*, 2001.

[6] D. Comaniciu, V. Ramesh and P. Meer. Real-time Tracking of Non-rigid Objects Using Mean Shift. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 142-149, 2000.

[7] S.Z. Li, L. Xhu, Z. Zhang, A. Blake, H. Zhang and H. Shum. Statistical Learning of Multi-view Face Detection. In *Proc. 7th European Conference on Computer Vision*, Copenhagen, Denmark, May 2002.

[8] M. Yang, D.J. Kriegman and N. Ahuja. Detecting Face in Images: A Survey. *IEEE Trans. on PAMI*, 24(1): 34-58, 2002.

[9] 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*, 55(1):119-139, 1997.

[10] A.S. Georghiades, P.N. Belhumeur and D.J. Kriegman. From Few To Many: Generative Models For Recognition Under Variable Pose and Illumination. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 277-284, 2000.

[11] K. Nummiaro, E. Koller-meier and L.Van Gool. An Adaptive Color-based Particle Filter. *Image and Vision Computing*, 21(1):99-110, 2003.