Novel Adaptive Eye Detection and Tracking for Challenging Lighting Conditions

Mahdi Rezaei and Reinhard Klette

The *.enpeda..* Project, Tamaki Campus
The University of Auckland, New Zealand

Abstract. The paper develops a novel technique that significantly improves the performance of Haar-like feature-based object detectors in terms of speed, detection rate under difficult lighting conditions, and reduced number of false-positives. The method is implemented and validated for driver monitoring under very dark, very bright, and normal conditions. The framework includes a fast adaptive detector designed to cope with rapid lighting variations, as well as an implementation of a Kalman filter for reducing the search region and indirect support of eye monitoring and tracking. The proposed methodology effectively works under low-light conditions without using infrared illumination or any other extra lighting support. Experimental results, performance evaluation, and comparing a standard Haar-like detector with the proposed adaptive eye detector, show noticeable improvements.

1 Introduction

Since the early 2000s, researchers such as Viola and Jones [13], Jesorsky et al. [7], or Hsu et al. [6] made important progress in model- and learning-based object detection methods. Despite of general improvements in detection methods, face and especially eye detection under non-ideal lighting conditions still requires further improvements. Even in recent efforts such as [10,12,15], limited verification tests have been applied for normal situations only. Driver-behaviour monitoring is an example for a challenging environment for eye analysis, where the light source is not uniform, symmetric, or the light intensity may rapidly and repeatedly change (e.g. due to entering a tunnel, shadow, turning into very bright light, or even sun strike). Although recent techniques for frontal face detection under normal lighting conditions are quite robust and precise [9,15,14,18], sophisticated and sensitive tasks such as driver eye-status monitoring (open, closed) and gaze analysis, are still far away from being solved accurately.

Among related work, there are publications on single and multi-classifier approaches for the addressed area of applications. Brandt et al. [2] designed a coarse-to-fine approach for face and eye detection using Haar wavelets to measure driver blinking with satisfactory results under ideal conditions. Majumdar [10] introduced a hybrid approach to detect facial features using Haar-like classifiers in HSV colour space but tested it on a very limited number of frontal faces only. Zhua and Ji [17] introduced robust eye detection and tracking under

J.-I. Park and J. Kim (Eds.): ACCV 2012 Workshops, Part II, LNCS 7729, pp. 427–440, 2013.

variable lighting conditions; however, their method is not effective without support of IR illumination. Research results in the field often suffer from a lack of verification and performance analysis on a wide-range of video or image data.

In this paper, we pursue four goals, (1) to improve noisy measurements of a Haar-classifier by a more stable solution for detecting and localizing features in the image plane, (2) to reduce the total computational cost by minimizing the search region based on a Kalman-filter face tracker, therefore indirectly reducing the "eye status" detection cost, (3) to minimize false detections by having a limited operational area within the image plane, and most importantly, (4) to overcome issues of eye-detection failures due to sophisticated lighting conditions, by introducing a novel technique, an *adaptive Haar detector*.

Section 2 outlines the main idea. Section 3 discusses our adaptive detection method. Section 4 informs about the implementation of the tracking module along with technical considerations for face and eye tracking. Section 5 discusses details of experimental and validation results with a performance analysis, comparing our method with the Viola-Jones method. Section 6 concludes.

2 Main Idea and Brief Methodology

Figure 1 illustrates the overall structure of our approach. Using Haar-feature based classifiers, two possible options are considered. In Option 1 we quantify the region of interest (ROI_1) as 100% for each classifier, which means that the whole area of an input image needs to be searched, from top left to the bottom right. Generally, such a full search needs to be repeated three times via three individual Haar classifiers in order to detect "face", "open eyes", or "closed eye" status. However, utilizing ground truth information for eye localization on two standard databases of FERET [3] and YALE [16], we estimated the eye location in a range of 0.55-0.75 as shown in Fig. 2. In case of head tilt, the eye location may vary in a range of 0.35-0.95 on one side of the face. Therefore, assuming an already detected face, an eye classifier can perform on regions A and B only (ROI_2), which are only 5.2% of the input image. If face detection fails, then a second full search in the image plane is required for eye classification (ROI_3), as we do not have any prior estimation for face location. This causes a total search cost of 200%. If the open-eye classifier detects at least one eye in segment A within the last five frames, then the closed-eye classifier is called to look at region C (ROI_4). This region covers about 3% of an VGA image. In brief, assuming an already detected face in the first stage, we have a maximum search area of 108.2% ($\mathrm{ROI}_1 + \mathrm{ROI}_2 + \mathrm{ROI}_4$) for complete eye status analysis, while a face detection failure leads to a search cost of 203% ($\mathrm{ROI}_1 + \mathrm{ROI}_3 + \mathrm{ROI}_4$).

Assessing 1,000 recorded frames and ground truth faces, we measured the mean size of the detected faces as 185 × 185 pixels which covers only 11% of a VGA image plane. Based on this idea we plan for a partial search that potentially defines a limited search region as Option 2 instead of Option 1.

As shown in Fig. 1, implementing a face tracker reduces the search region for face detection, thus a faster eye analyses through the tracked face region. Later we discuss that using a tracking solution as Option 2, the total search cost can be reduced to around 34.6% (ROI_{1_1} + ROI_2 + ROI_4) which is at least 6 times faster than Option 1.

In addition to an optimized search policy, we require proper face localization from the first step, followed by robust eye status detectors to be applicable for all lighting conditions. Figure 3 shows examples that point to a need for robust eye detectors to be adaptive under extremely challenging lighting conditions. This requirement is considered in the left feedback cycle in Fig. 1, called classifier tuning and adaptation phase, which introduces adaptive Haar-like eye detection to overcome the weakness of a standard Viola-Jones detector [13].

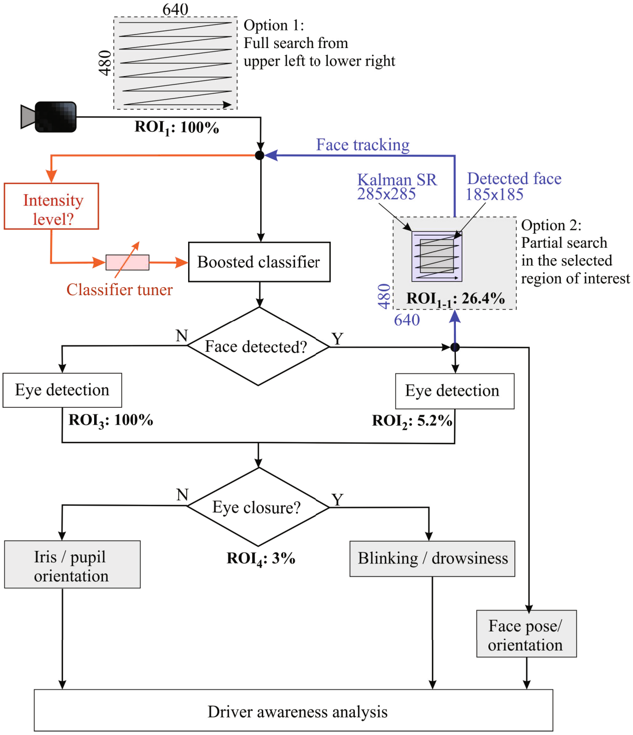

Fig. 1. Driver awareness monitoring: brief flowchart

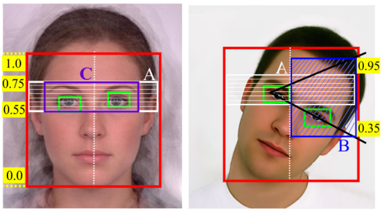

Fig. 2. Optimized ROIs for eye tracking, after successful face detection

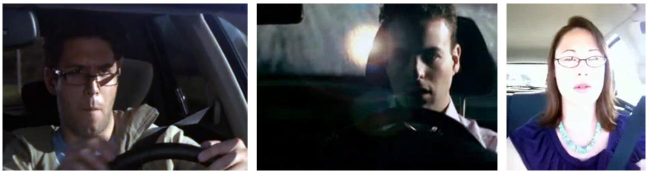

Fig. 3. Examples of "unusual" lighting conditions while driving, causing difficulties for driver's eye monitoring (images in the public domain)

3 Adaptive Classification and Detection

The adaptation module is detailed in three sub-sections, addressing the recognition of weakness of a Viola-Jones detector [13], statistical analysis of intensity changes around the eye region, and dynamic parameter adaptation to overcome inefficiency of Haar-feature based detectors under non-ideal conditions.

3.1 Weakness of Viola-Jones for Challenging Lighting Conditions

We tried five well-recognized and publicly available [11] Haar classifiers developed by Castrillon, Lienhart, Yu, and Hameed, in our nominated application, driver monitoring. Although they are quite robust for non-challenging and normal lighting scenes, we realized that due to frequent shadows and artificial lighting in day and night, those Haar-like classifiers are likely to fail. The situation becomes even more complicated when a part of the driver's face is brighter than the other part (due to light falling in through a side-window), making eye status detection extremely difficult. We also compiled and trained our own classifier based on a large dataset of +12,000 positive images from YALE [16], FERET [3], BioID [1], and FTD [5]. Applying an AdaBoost machine learning technique [4], we obtained best results for our trained Haar-like classifier with the following parameters:

- Positive images of size 21 × 21 pixel
- Number of weak classifiers (stages): 15
- Minimum expected hit rate for each stage: 99.8%
- Maximum acceptable false alarm at each stage: 40%
- Trimming threshold: 0.95

After training and creation of the classifier, we had to utilize the classifier in the real-world with parameters which are normally similar to the trained parameters. Main parameters are:

- Initial search window size (SWS) which should normally be the same as the scale size of positive images (i.e. 21 × 21 as above)
- Scale factor (SF) to increase the SWS in each subsequent search iteration (e.g. 1.2 which means 20% increase in window size for each search iteration)

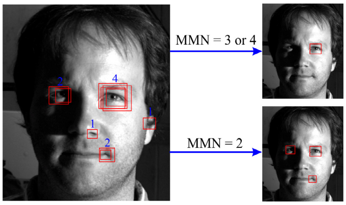

Fig. 4. Left: Initial results of eye detection. Right: Finally detected eyes after trimming by different MNN factors.

- Minimum expected number of detected neighbours (MNN) which is needed to confirm an object, when there are multiple object candidates in a small region (e.g. 3)

In general, a smaller SF means a more detailed search in each iteration, but it also causes higher computational costs.

Decreasing MNN, causes increase of detection rate; however, it increase false detection rate as well. Larger values for MNN lead to more strictness to confirm a candidate for face or eye, thus a reduced detection rate.

Figure 4 shows potential issues related to the MNN parameter in Viola-Jones techniques. The figure shows 10 initial eye candidates before trimming by the MNN parameter. Detections are distributed in 5 regions, each region shows 1 to 4 overlapping candidates. In order to minimize this problem, it is common to assign a trade-off value for the MNN parameter to gain the best possible results. Figure 4, top right, shows one missed detection with MMN equals 3 or 4, and Fig. 4, bottom right, shows one false detection with MMN equals 2. MMN equals 1 causes 3 false detections, and any MMN greater than 4 will lead to no detection at all; so there is no optimum MNN value for this example.

We conclude that although we can define trade-off values for SWS, SF, and MNN to obtain the optimum detection rate for "ideal" video sequences, however, a diversity of changes in light intensity over the target object can still significantly affect the performance of the given classifier in terms of TP and FP rates.

3.2 Hybrid Intensity Averaging

To cope with the above mentioned issues, we propose that Haar-classifier parameters have to be adaptive, varying with time depending on lighting changes. Figures 5 and 6 illustrate that we cannot measure illumination changes by simple intensity averaging over the input frame: In the driving application, there can be strong back-light from the back windshield, white pixel values around the driver's cheek or forehead (due to light reflections), or dark shadows on the driver's face. All these conditions may negatively affect the overall mean intensity measurement. Analysing various recorded sequences, we realized that pixel intensities around eyes can change independently from surrounding regions in the input sequence. Focusing on a detected face region, Fig. 5, right, shows very dark and very bright spots in two sample faces (in a grey-level range of 0-35

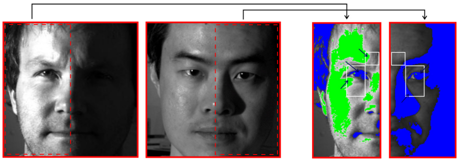

Fig. 5. Mean intensity measurement for eye detection, by excluding *very bright* and *very dark* regions. Green: thresholding range 210-255. Blue: thresholding range 0-35.

and 210-255, respectively). It also shows a considerable illumination difference for the left and right side of a driver's face. Apart from the iris intensity (for dark or light eyes), the surrounding eye intensities play a very crucial role in eye detection. Thus, proper classifier parameter adjustment based on proper intensity measurement in the region surrounding an eye can guaranty robust eye detection. Following this consideration, we defined white rectangles around eyes (Fig. 5, right) which can not only provide a good approximation of both vertical and horizontal light intensities around the eyes, but they are also very marginally influenced by green or blue (very bright or very dark) regions.

Considering an already detected face, and expected eye regions A and C based on Fig. 2, we can geometrically define C_r, F_r, C_l, and F_l as being the optimum regions in order to gain an accurate estimation of light intensity around the eyes (see Fig. 6). We also consider independent classifier parameters for the left and right half of the face, as each half of the face may receive different and non-uniform light exposures.

Performing a further analytical step, Fig. 5, right, shows that a few small green or blue segments (extreme dark or light segments) have entered the regions of white rectangles. This can affect the actual mean intensity calculations in the C or F regions. Thus, in order to reduce the affect of this kind of noise into our measurements, we apply a hybrid averaging by combining *mean* and *mode* (Mo) of pixel intensities as follows:

$$I_r(\alpha) = \frac{1}{2} \cdot \left[\alpha \cdot \mathtt{Mo}(C_r) + \frac{(1-\alpha)}{m} \sum_{i=1}^{m} C_{r_i} + \alpha \cdot \mathtt{Mo}(F_r) + \frac{(1-\alpha)}{n} \sum_{j=1}^{n} F_{r_j} \right] \quad (1)$$

where $I_r(\alpha)$ is the hybrid intensity value of the *right eye* region of the face, m and n are the total numbers of pixels in C_r and F_r regions; C_r and F_r are in $[0, 255]$ point to cheek and forehead light intensity, respectively.

An α-value of 0.66 assumes a double importance of *mode* intensity measurement compared to *mean* intensity; Integration of *mode* reduces the impact of eye iris colour (i.e. blue segments) as well as of very bright pixels (i.e. green segments) for our adaptive intensity measurement. Similarly, we can calculate $I_l(\alpha)$ as hybrid intensity value of *left eye* region.

3.3 Parameter Adaptation

The final step of the detection phase is *classifier parameter adjustment* based on the measured I_r and I_l values, to make our classifier adaptive for every

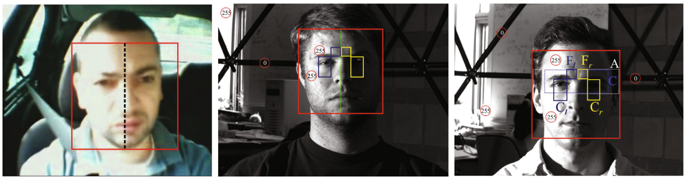

Fig. 6. Selected regions to sum up the mean intensity around eyes. Images Source: Yale database.

Table 1. Optimum parameters for 10 selected intensity levels in terms of *Search Window Size*, *Scale Factor*, and *Minimum Number of Neighbours*. FS: detected *Face Size*.

Light Intensity	*SWS*	*SF*	*MNN*
0	FS/5.0	1.10	1
20	FS/4.5	1.12	2
50	FS/3.5	1.30	9
75	FS/4.0	1.15	7
90	FS/4.0	1.30	10
120	FS/4.2	1.25	16
155	FS/5.0	1.35	15
190	FS/4.5	1.30	14
220	FS/4.6	1.25	9
255	FS/4.0	1.35	7

single input frame. Now we need to find optimum parameters (SWS, SF, MNN) for all the intensity ranges between 0 to 255, which is a highly time-consuming practical tasks. Instead, we defined optimum parameters for 10 selected intensities, followed by a data interpolation method to extend those parameters to all intensity ranges.

Table 1 shows optimum parameter values for 10 data points obtained from 20 recorded videos in different weather and lighting conditions. These factors are adjusted to lead to the highest rate of true positives for each of the 10 given intensities. The parameter values in Table 1 show a non-linear behaviour over intensity changes; therefore we apply non-linear *cubic interpolation* and *Lagrange interpolation* to extend adapted values to an intensity range from 0 to 255.

4 Tracking and Search Minimization

This section pursues the goals of minimizing the search region for eye status detection, time efficiency, less computational cost, more precise detections, and lower rate of false detection.

4.1 Tracking Considerations

A simple tracking around the previously detected face can easily fail due to a fast change in both face size and moving trajectory (Fig. 7). Therefore we need to perform a dynamic and intelligent tracking strategy to minimize the search region. We apply a Kalman filter.

Figure 8 shows the brief structure of Kalman filter [8] including time update and measurement steps, where $\hat{x}_k^-$ and $\hat{x}_k^+$ are priori and posteriori states estimated for centre of the detected face, z_k is a *Haar-classifier measurement vector*, A is an $n \times n$ matrix referred to as *state transition matrix* which transforms the previous state at time step $k-1$ into the current state at time step k, B is an $n \times l$ matrix referred to as *control input transition matrix* which relates to optional control parameter $u \in \Re^l$, w_k is process noise which is assumed to be Gaussian and white, and H is an $m \times n$ matrix called *measurement transition matrix* which relates to state to the measurement. P_k^- and P_k^+ are the *priori and posteriori estimation error covariance* based on predicted and measured values (by Haar-classifier), and K_k is the *Kalman gain.*

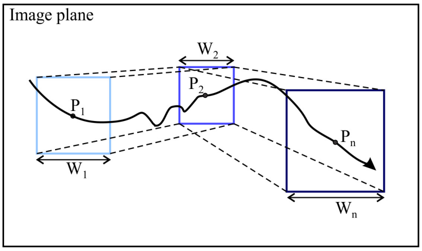

Fig. 7. Sample movement trajectory: simultaneous changes in face position and size

4.2 Filter Modelling and Implementation

There are many different motion equations such as linear acceleration, circular acceleration, or Newton mechanics; however, for a driver's face movements we simply consider a linear dynamic system and assume constant acceleration in a short Δt time frame. We implemented and modelled the filter with the following elements. First, the state vector is defined as $x_t = [x \quad y \quad w \quad v_x \quad v_y \quad a_x \quad a_y]^T$. Also based on the theory of motion we have that

$$p(t+1) = p(t) + v(t)\Delta t + a(t)\frac{\Delta t^2}{2} \tag{2}$$

$$v(t+1) = v(t) + a(t)\Delta t \tag{3}$$

where p, v, a, Δt are position, velocity, acceleration, and time difference between input images, respectively. Δt is the average processing time which is the time between starting the process of face detection on a given frame at time k until end of the eye detection process and accepting the next frame at time $k+1$.

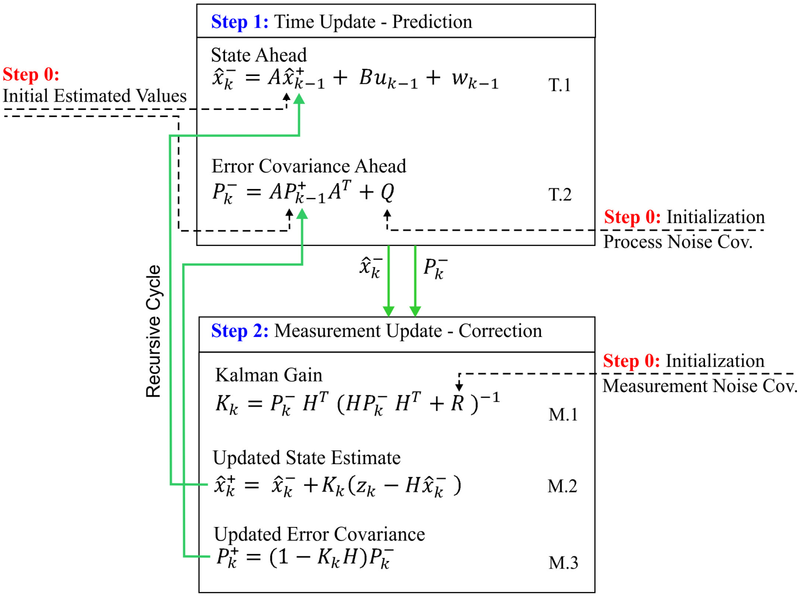
Step 1: Time Update - Prediction
State Ahead
$\hat{x}_k^- = A\hat{x}_{k-1}^+ + Bu_{k-1} + w_{k-1}$
T.1
Error Covariance Ahead
$P_k^- = AP_{k-1}^+A^T + Q$
T.2
Step 0:
Initial Estimated Values
Step 0: Initialization
Process Noise Cov.
$\hat{x}_k^-$
P_k^-
Recursive Cycle
Step 2: Measurement Update - Correction
Kalman Gain
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
M.1
Step 0: Initialization
Measurement Noise Cov.
Updated State Estimate
$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
M.2
Updated Error Covariance
$P_k^+ = (1 - K_kH)P_k^-$
M.3

Fig. 8. Kalman filter in a nutshell

Relying on the definition of state transition in equation T.1, we modelled the transition matrix A as a 7×7 matrix. Thus, the next state is estimated as below:

$$\begin{pmatrix} x_k \\ y_k \\ w_k \\ v_{x_k} \\ v_{y_k} \\ a_{x_k} \\ a_{y_k} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & \frac{1}{2}\Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ w_{k-1} \\ v_{x_{k-1}} \\ v_{y_{k-1}} \\ a_{x_{k-1}} \\ a_{y_{k-1}} \end{pmatrix} + \begin{pmatrix} rand(2e-3) \\ rand(2e-3) \\ . \\ . \\ . \\ . \\ rand(2e-3) \end{pmatrix}$$

Before running the filter in the real world, we need to have a few initializations including $\hat{x}_0^+$, $\hat{P}_0^+$, R, Q, and H. R directly depends on measurement accuracy of our camera along with the accuracy of our face/eye detection algorithm. Comparing the ground truth information and the result of our face detection method, we determined the variance of the measurement noise in our system as $R = rand(1e-4)$. The determination of Q is generally more difficult than of R, and needs to be tuned manually. A good tuning of R and Q stabilizes K_k and P_k very quickly after a few iteration of filter recursion. We got the best system stability when we set the process noise to $Q = rand(2e-3)$.

We take $H = 1$ because the measurement is composed of only the state value and some noise. Matrix B is omitted as there is no external control for driver face movement. For $\hat{x}_0^+$ and $\hat{P}_0^+$ we assumed the initial position of a face at position $x = 0$ and $y = 0$, with an initial speed of zero, as well as posteriori error covariance of 0. We also considered Δt as being between 33 and $170ms$, based on computation cost and a processing rate between 6 to $30Hz$.

5 Experiments and Validation

A grey-level VGA camera at a distance of $60cm$ to the driver seat is used to take continuous recording of the driver seat area. Figures 9-11 show results of eye status detection before and after implementation of the adaptation module. The images have been selected from 20 recorded video sequences with extremely varying lighting. Table 2 provides details of TP and FP detection rates performed on 2 sample videos (5 minutes each) and 2 face datasets (2,000 images each).

Figure 12 illustrates partial face tracking results and error indices in x-coordinates, for 450 recorded sequences while driving. Using adaptive Haar-like detectors, we rarely faced detection failure for more than 5 continued frames; however, we intentionally deactivated the detection module for up to 15 continued frames to see the tracking robustness (grey bars, Fig. 12). Results show promising tracking without any tracking divergence. Similar results were obtained for y-coordinates and face size tracking. Comparing ground truth and tracking results, the average error index was ± 0.04. Adding a safety margin of 4% around the tracking results, we can recursively define an optimized rectangular search region for the adaptive Haar-classifier instead of a blind search in the whole image plane. We define $P1_k = (x1_{k-1} - 0.04 \times 640,\, y1_{k-1} - 0.04 \times 480)$ and $P2_k = (x2_{k-1} + 0.04 \times 640 + 1.4w,\, y2_{k-1} + 0.04 \times 480 + 1.4w)$ as the optimized search region, where $P1_k$ and $P2_k$ point to the upper-left and lower-right corners of search regions at time k; pairs of $(x1_{k-1},\, y1_{k-1})$ and $(x2_{k-1},\, y2_{k-1})$ are upper-left and lower-right coordinates of predicted faces at time $k - 1$; w is the predicted face width at time $k - 1$.

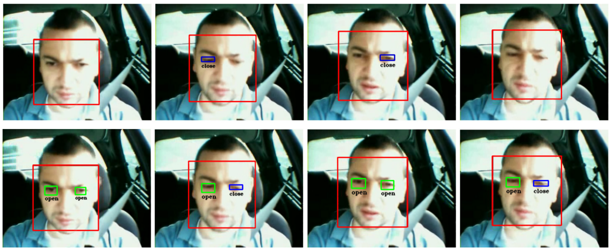

Fig. 9. Video 1: Face and eye detection under sun-strike; standard classifier (top) vs. adaptive classifier (bottom)

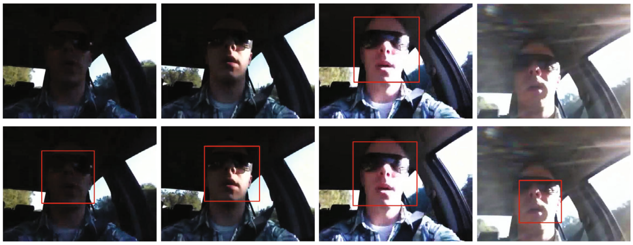

Fig. 10. Video 2: Face detection with sunglasses under light-varying conditions; standard classifier (top) vs. adaptive classifier (bottom)

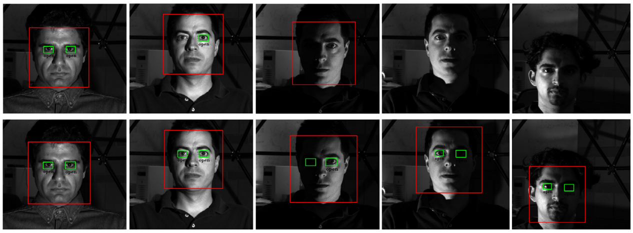

Fig. 11. Face and eye detection under difficult lighting conditions; standard classifier (top) vs. adaptive classifier (bottom). Image source: Yale database.

Table 2. Performance analysis for *standard* and *adaptive classifier*

	Standard V-J Classifier						Proposed Adaptive Classifier					
	Face		Open		Closed		Face		Open		Closed	
	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
Video 1	97.5	0.01	82	3.2	86	4.4	99.3	0	96.1	0.27	95.7	0.32
Video 2	81.1	1.02	-	0.5	-	0.32	94.6	0.01	-	0.01	-	0
Yale DB	86.3	0.05	79.4	0.1	-	0.07	98.8	0.02	97.3	0	-	0.01
Closed Eye DB	92.2	0.06	87.5	3.7	84.2	3.9	99.5	0.02	99.2	0.4	96.2	0.18

Figure 13.*a* shows good tracking after 5 frames, and 13.*b* shows perfect tracking after 10 frames. Figure 13.c displays failed face detection due to face occlusion while steering; however, successful face tracking (yellow frame) lead to proper eye detection. Figure 13.d shows another good "match" of detection and tracking with accurate closed-eye detection. Figure 14 shows very good results for eye detection and tracking at night when having sharp back-lights, strong shades, and very dark conditions.

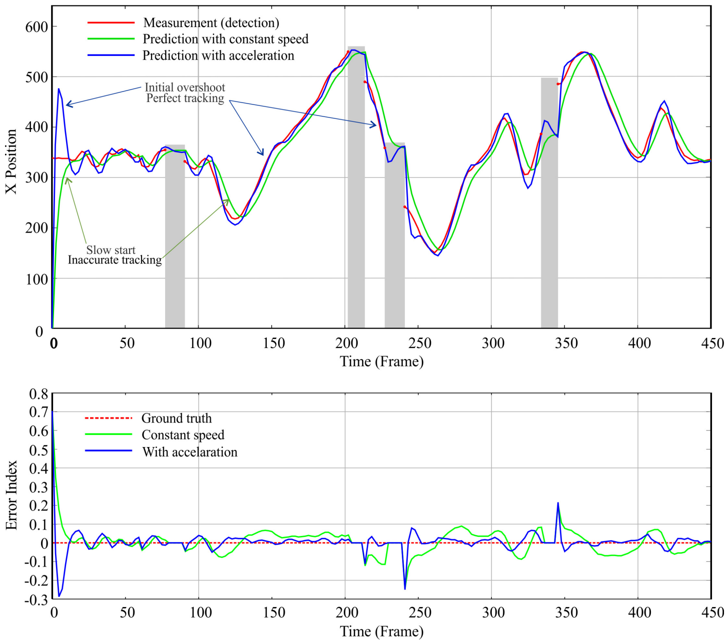

Fig. 12. Tracking results with and without accelerations. Grey blocks: No detections, but still good tracking results.

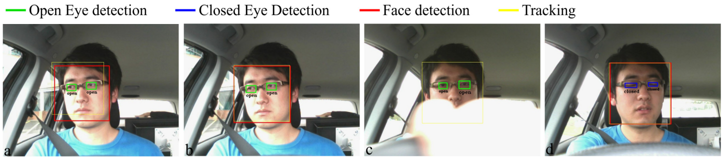

Fig. 13. *Face*, *open eye*, and *closed eye* detection and tracking while driving in daylight.

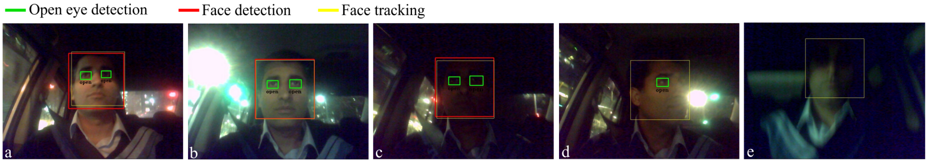

Fig. 14. *Face*, *open eye*, and *closed eye* detection and tracking at night under difficult lighting. (e) detection failure due to motion blur; however still robust tracking.

6 Conclusions

The paper introduced a fast and effective detection framework that enables accurate facial feature tracking under difficult lighting, performing clearly better than standard Viola-Jones classifiers. While the *tracking module* minimized the search region, the *adaptive module* focused on the given limited region to adapt the SWS, SF, MNN parameters depending on pixel by pixel intensity changes in eye-pair surrounding areas. Both modules recursively boosted each other; the face tracking module provides an optimum search area for eye detection, and in turn, the adaptive eye detection module provides geometrical face location estimation to support a Kalman tracker in case of tracking failure. We gained six times faster processing and more accurate results using only a low-resolution VGA camera, without any application of IR light, or any preprocessing or illumination normalization techniques. Our solution is suggested to be an amendment for any kind of a Haar-like classifier, to improve in challenging environments. For future work we suggest integration of pre-processing techniques and a comparison with illumination invariant methods such as SIFT or LBP.

References

1. BioID Database (2012), `http://www.bioid.com/downloads/software/bioid-face-database`
2. Brandt, T., Stemmer, R., Rakotonirainy, A.: Affordable visual driver monitoring system for fatigue and monotony. Systems Man Cybernetics 7, 6451–6457 (2004)
3. FERET Face Database (2012), `http://www.itl.nist.gov/iad/humanid/feret/`
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Computer System Sciences 55, 119–139 (1997)
5. Face of tomorrow database (2012), `http://www.faceoftomorrow.com/posters.asp`
6. Hsu, R., Abdel-Mottaleb, M., Jain, A.K.: Face detection in color images. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 696–706 (2002)
7. Jesorsky, O., Kirchberg, K.J., Frischholz, R.W.: Robust Face Detection Using the Hausdorff Distance. In: Bigun, J., Smeraldi, F. (eds.) AVBPA 2001. LNCS, vol. 2091, pp. 90–95. Springer, Heidelberg (2001)
8. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Engineering 82, 35–45 (1960)
9. Li, S.Z., Jian, A.K.: Handbook of Face Recognition. Springer, London (2011)
10. Majumdar, A.: Automatic and robust detection of facial features in frontal face images. In: Proc. Int. Conf. Computer Modelling Simulation, pp. 331–336 (2011)
11. World recognized Haar classifier contributors for face and eye detection (2012), `http://opencv.willowgarage.com/wiki/Contributors`
12. Tsao, W., Lee, A.J.T., Liu, Y., Chang, T., Lin, H.: A data mining approach to face detection. Pattern Recognition 43, 1039–1049 (2010)
13. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR, pp. 511–518 (2001)
14. Wu, Y., Ai, X.: Face detection in color images using AdaBoost algorithm based on skin color information. In: Proc. Int. Workshop on Knowledge Discovery Data Mining, pp. 339–342 (2004)

15. Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast asymmetric learning for cascade face detection. IEEE Trans. Pattern Analysis and Machine Intelligence 30, 369–382 (2008)
16. YALE Face Database (2012), `http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html`
17. Zhu, Z., Ji, Q.: Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. J. Computer Vision Image Understanding 98, 124–154 (2005)
18. Zhang, X., Gao, Y.: Face recognition across pose: A review. Pattern Recognition 42, 2876–2896 (2009)