Face Tracking with Automatic Model Construction

Jesus Nuevo, Luis M. Bergasa*, David F. Llorca, Manuel Ocaña

Department of Electronics, Universidad de Alcala. Esc. Politecnica, Crta Madrid-Barcelona, Km 33,600. 28871 Alcala de Henares, Madrid

Abstract

~~Driver inattention is one of the major causes of traffic crashes, claiming thousands of lives every year. Face tracking is one of the first stages in safety systems that relay on computer vision to detect inattention.~~ This paper describes an active model with a robust texture model built on-line. The model uses one camera and it is able to operate without active illumination. The texture model is defined by a series of clusters, which are built in a video sequence using previously encountered samples. This model is used to search for the corresponding element in the following frames. An on-line clustering method, named *leaderP* is described and evaluated on an application of face tracking. A 20-point shape model is used. This model is built offline, and a robust fitting fuction is used to restrict the position of the points. *Our proposal is to serve as one of the stages in a driver monitoring system.* To test it, a new set of sequences of drivers recorded outdoors and in a realistic simulator has been compiled. Experimental results for typical outdoor driving scenarios, with frequent head movement, turns and occlusions are presented. Our approach is tested and compared with the Simultaneous Modeling and Tracking (SMAT) [1], and the recently presented Stacked Trimmed Active Shape Model (STASM) [2], and shows better results than SMAT and similar fitting error levels to STASM, with much faster execution times and improved robustness.

Keywords: Face tracking, appearance modeling, incremental clustering, robust fitting, driver monitoring

1. Introduction

Driver inattention is a major cause of traffic accidents, and it has been found to be involved in some form in 80 percent of the crashes and 65 percent of the near crashes within 3 seconds of the event [3]. Monitoring a driver to detect inattention is a complex problem that involves physiological and behavioural elements. Different works have been presented in recent years, focused mainly in drowsiness, with a broad range of techniques. Physiological

*Corresponding author: Tel. (+34) 91885 6569

Email addresses: `jnuevo@depeca.uah.es` (Jesus Nuevo), `bergasa@depeca.uah.es` (Luis M. Bergasa), `llorca@depeca.uah.es` (David F. Llorca), `mocana@depeca.uah.es` (Manuel Ocaña)

measurements such as electro-encephalography (EEG) [4] or electro-oculography (EOG), provide the best data for detection [4]. The problem with these techniques is that they are intrusive to the subject. Moreover, medical equipment is always expensive.

Lateral position of the vehicle inside the lane, steering wheel movements and time-to-line crossing are commonly used, and some commercial systems have been developed [5, 6]. These techniques are not invasive, and to date they obtain the most reliable results. However, the measurements they use may not reflect behaviors such as the so-called micro-sleeps [7]. They also require a training period for each person, and thus are not applicable to the occasional driver.

Drivers in fatigue exhibit changes in the way their eyes perform some actions, like moving or blinking. These actions are known as *visual behaviors*, and are readily observable in drowsy and dristracted drivers. Face pose [8] and gaze direction also contain information and have been used as another element of inattention detection systems [9]. Computer vision has been the tool of choice for many researchers to be used to monitor visual behaviours, as it is non-intrusive. Most systems use one or two cameras to track the head and eyes of the subject [10, 11, 12, 13, 14]. A few companies commercialize systems [15, 16] as accessories for installation in vehicles. These systems require user-specific calibration, and some of them use near-IR lighting, which is known to produce eye fatigue. Reliability of these systems is still not high enough for car companies to take on the responsibility of its production and possible liability in case of malfunctioning.

Face location and tracking are the first processing stages of most computer vision systems for driver monitoring. Some of the most successful systems to date use near-IR active illumination [17, 18, 19], to simplify the detection of the eyes thanks to the *bright pupil* effect. Near-IR illumination is not as useful during the day because sunlight also has a near-IR component. As mentioned above, near-IR can produce eye fatigue and thus limits the amount of time these systems can be used on a person.

Given the complexity of the problem, it has been divided in parts and in this work only the problem of face tracking is addressed.

This paper presents a new active model with the texture model built incrementally. We use it to characterize and track the face in video sequences. The tracker can operate without active illumination. The texture model of the face is created online, and thus specific for each person without requiring a training phase. A new online clustering algorithm is described, and its performance compared with the method proposed in [1]. Two shape models, trained online and off-line, are compared. This paper also presents a new video sequence database, recorded in a car moving outdoors and in a simulator. The database is used to assess the performance of the proposed face tracking method in the challenging environment a driver monitoring application would meet. *No evaluations of face pose estimation and driver inattention detection are performed.*

The rest of the paper is structured as follows. Section 2 presents a few remarkable works in face tracking in the literature that are related to our proposal. Section 3 describes our approach. Section 4 describes the video dataset used for performance evaluation, and experimental results. This paper closes with conclusions and future work.

2. Background

Human face tracking is a broad field in computing research [20], and a myriad of techniques have been developed in the last decades. It is of the greatest interest, as vast amounts of information are contained in face features, movements and gestures, which are constantly used for human communication. Systems that work on such data often use face tracking [21, 22].

Non-rigid object tracking has been a major focus of research in latter years, and general purpose template-based trackers have been used to track faces in the literature with success. Several efficient approaches have been presented [23, 24, 25, 26].

Statistical models have been used for face modeling and tracking. Active Shape Models [27] (ASM) are similar to the active contours (*snakes*), but include constraints from a Point Distribution Model (PDM) [28] computed in advance from a training set. Advances in late years have increased their robustness and precision to remarkable levels (STASM,[2]). Extensions of ASM that include modeling of texture have been presented, of which Active Appearance Models (AAMs) [29] are arguably the best known. Active Appearance Models are global models in the sense that the minimization is performed over all pixels that fall inside the mesh defined by the mean of the PDM. All these models have an offline training phase, which require comprehensive training sets so they can generalize properly to unseen instances of the object. This is time consuming process, and there is still the risk that perfectly valid instances of the object would not be modeled correctly.

Several methods that work without *a priori* models have been presented in the literature. Most of them focus on patch tracking on a video sequence. The classic approach is to use the image patch extracted on the first frame of the sequence to search for similar patches on the following frames. Lukas-Kanade method [30] was one of the first proposed solutions and it is still widely used. Jepson *et al.* [31] presented a system with appearance model based on three components: a stable component that is learned over a long period based on wavelets, a 2-frame tracker and an outlier rejection process. Yin and Collins [32] build an adaptive view-dependent appearance model on-line. The model is made of patches selected around Harris corners. Model and target patches are matched using correlation, and the change in position, rotation and scale is obtained with the Procrustes algorithm.

Another successful line of work in object tracking without *a priori* training is based on classification instead of modeling. Collins and Liu [33] presented a system based on background/foreground discrimination. Avidan [34] presents one of the many systems that use machine learning to classify patches [35, 36]. Avidan uses weak classifiers trained every frame and AdaBoost to combine them. Pilet *et al.* [37] train keypoint classifiers using Random Trees that are able to recognize hundreds of keypoints in real-time.

Simultaneous Modeling and Tracking (SMAT) [1] is in line with methods like Lucas-Kanade, relaying on matching to track patches. Lukas-Kanade extracts a template at the beginning of the sequence and uses it for tracking, and will fail if the appearance of the patch changes considerably. Matthews *et al.* [38] proposed an *strategic update* of the template, which keeps the template from the first frame to correct errors that appear in the localization. When the error is too high, the update is blocked. In [39], a solution is proposed with fixed template that adaptively detected and selected the window around the features. SMAT builds a more complex model based on incremental clustering.

In this paper we combine concepts from active models with the incremental clustering proposed in SMAT. The texture model is created online, making the model adaptative, while the shape model is learnt offline. The clustering used by SMAT has some limitations, and we propose some modifications to obtain a more robust model and better tracking. We name the approach *Robust SMAT* for this reason.

Evaluation of face tracking methods is performed in most works with images captured indoors. Some authors use freely available image sets, but most of them test on internal datasets created by them, which limits the validity of a comparison with other systems. Only a few authors [40][41] have used images recorded in a vehicle, but the number of samples is limited. To the best of our knowledge, there is no publicly available video dataset of people driving, either in a simulator or in a real road. We propose a new dataset that covers such scenarios.

3. Robust Simultaneous Modeling and Tracking

This section describes the Simultaneous Modeling and Tracking (SMAT) of Dowson and Bowden [1], and some modifications we propose to improve its performance. SMAT tries to build a model of appearance of features and how their positions are related (the structure model, or *shape*), from samples of texture and shape obtained in previous frames.

The models of appearance and shape are independent. Fitting is performed in the same fashion of ASM: the features are first found separatedly using correlation, and then their final positions are constrained by the shape model. If the final positions are found to be reliable and not caused by fitting errors, the appearance model is updated, otherwise it is left unchanged. Figure 1 shows a flow chart of the algorithm.

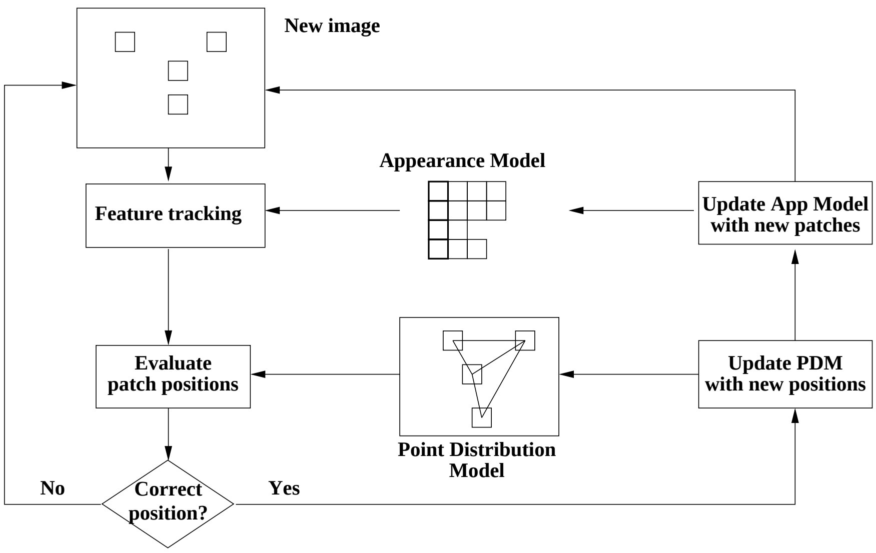


Figure 1: SMAT block diagram

3.1. Appearance modeling

Each one of the possible appearances of an object, or a feature of it, can be considered as a point in a feature space. Similar appearances will be close in this space, away from other points representing dissimilar appearances of the object. These groups of points, or clusters, form a mixture model that can be used to define the appearance of the object.

SMAT builds a library of exemplars obtained from previous frames, image patches in this case. Dowson and Bowden defined a series of clusters by their median patch, also known as *representative*, and their variance. A new incoming patch is made part of the cluster if the distance between it and the median of the cluster is below a threshold that is a funcion of the variance. The median and variance of a cluster are recalculated everytime a patch is added to it. Up to M exemplars per cluster are kept. If the size limit is reached, the most distant element from the representative is removed.

Everytime a cluster is updated, the weight of the clusters is recalculated as in equation 1:

$$w_k^{(t+1)} = \begin{cases} (w_k^{(t)} + \alpha)\frac{1}{1+\alpha} & if \quad k = k_u \\ w_k^{(t)}\frac{1}{1+\alpha} & \text{otherwise} \end{cases} \tag{1}$$

where $\alpha \in [0, 1)$ is the learning rate, and k_u is the index of the updated cluster. The number of clusters is also limited to K. If K is reached, the cluster with the lowest weight is discarded.

In a later work, Dowson *et al.* [42], introduced a different condition for membership, that compares the probability of the exemplar belonging to foreground (a cluster) or to the background

$$\frac{p(fg\,|d(x, \mu_n), \sigma_{fg_n})}{p(bg\,|d(x, \mu_n), \sigma_{bg_n})} \tag{2}$$

where σ_{fg_n} is obtained from the distances between the representative and the other exemplars in the cluster, and σ_{bg_n} is obtained from the distances between the representative and the exemplars in the cluster offset by 1 pixel.

We have found that this clustering method can be improved in several ways. The adapting nature of the clusters could theoretically lead two or more clusters to overlap. However, in our tests we have observed that the opposite is much more frequent: the representative of the cluster rarely changes after the cluster has reached a certain number of elements.

Outliers can be introduced in the model in the event of an occussion of the face by a hand or other elements like a scarf. In most cases, these exemplars would be far away from the representative in the cluster. To remove them and reduce memory footprint, SMAT keeps up to M exemplars per cluster. If the size limit is reached, the most distant element from the representative is removed. When very similar patches are constantly introduced, one of them will be finally chosen as the median, and the variance will decrease, overfitting the cluster and discarding valuable exemplars. At a frame rate of 30 fps, with M set to 50, the cluster will overfit in less than 2 seconds. This would happen even if the exemplar to be removed is chosen randomly. This procedure will discard valuable information and future, subtle changes to the feature will lead to the creation of another cluster.

We propose an alternative clustering method, named *leaderP*, to partially solve these and other problems. The method is a modification of the *leader* algorithm [43, 44], arguably the simplest and most frequently used incremental clustering method. In *leader*, each cluster $\mathcal{C}_i$ is defined by only one exemplar, and a fixed membership threshold T. It starts by making the first exemplar the *representative* of a cluster. If an incoming exemplar fulfills being within the threshold T it is marked as member of that cluster, otherwise it becomes a cluster on its own. The pseudocode is shown in algorithm 1.

Algorithm 1 Leader clustering

Let $C = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ be a set of n clusters, with weights $\{w_1^t, \ldots, w_n^t\}$

procedure LEADER(E, C) ▷ cluster patch E
    for all $\mathcal{C}_i \in C$ **do**
        if $d(\mathcal{C}_k, E) < T$ **then** ▷ Check if patch $E \in \mathcal{C}_k$
            UPDATEWEIGHTS($w_1^t, \ldots, w_n^t$) ▷ As in equation 1
            return
        end if
    end for
    Create new cluster $\mathcal{C}_{n+1}$, with E as representative.
    Set $w_{n+1}^{t+1} \leftarrow 0$ ▷ Weight of new cluster $\mathcal{C}_{n+1}$
    $C \leftarrow C \cup \mathcal{C}_{n+1}$ ▷ Add new cluster to the model
    if $n + 1 > K$ **then** ▷ Remove the cluster with lowest weight
        Find $\mathcal{C}_k \mid w_k \leq w_i \quad i = 1, \ldots, n$
        $C \leftarrow C \setminus \mathcal{C}_k$
    end if
end procedure

On the other hand, *leaderP* keeps the first few exemplars added to the cluster are kept, up to P. The median of the cluster is chosen as the representative, as in the original clustering of Dowson and Bowden. When the number of exemplars in the cluster reaches P, all exemplars but the representative are discarded, and it starts to work under the leader algorithm. P is chosen as a small number (we use $P = 10$). The membership threshold is however flexible: the distances between the representative and each of the exemplars that are found to be members of the cluster is saved, and the variance of those distances is used to calculate the threshold. Because the representative is fixed and distance is a scalar, many values can be kept in memory without having a impact on the overall performance. Keeping more values reduces the risk of overfitting.

The original proposal of SMAT used Mutual Information (MI) as a distance measure to compare the image patches, and found it to perform better that Sum of Squared Differences (SSD), and slightly better than correlation in some tests. Any definition of distance could be used. We have also tested Zero-mean Normalized Cross-Correlation (ZNCC). Several types of warping were tested in [42]: translation, euclidean, similarity and affine. The results showed an increasing failure rate as the degrees of freedom of the warps increased. Based on this, we have chosen to use the simplest, and the patches are only translated depending on the point distribution model.

3.2. Shape model

In the original SMAT of Dowson and Bowden, the shape was also learned on-line. The same clustering algorithm was used, but the membership of a new shape to a cluster was calculated using Mahalanobis distance.

Our method relies on the pre-learned shape model. The restrictions on using a pre-learned model for shape are less than those for an appearance model, as it is of lower dimensionality and the deformations are easier to model. It has been shown [45] that location and tracking errors are mainly due to appearance, and that a generic shape model for faces is easier to construct. We use the method of classic ASM [27], which applies PCA to a set of samples created by hand and extracts the mean $\mathbf{s}_0$ and an orthogonal vector basis $(\mathbf{s}_1, \ldots, \mathbf{s}_N)$. The shapes are first normalized and aligned using Generalized Procrustes Analysis [46].

Let $\mathbf{s} = (x_0, y_0, \ldots, x_{n-1}, y_{n-1})$ be a shape. A shape can be generated from this base as

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{m} p_i \cdot \mathbf{s}_i \tag{3}$$

Using L_2 norm, the coefficients $\mathbf{p} = (p_1, \ldots, p_N)$ can be obtained for a given shape $\mathbf{s}$ as a projection of $\mathbf{s}$ on the vector basis

$$\mathbf{p} = \mathbf{S}^T(\mathbf{s} - \mathbf{s}_0), \quad p_i = (\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{s}_i \tag{4}$$

where $\mathbf{S}$ is a matrix with the eigenvectors $\mathbf{s}_i$ as rows. The estimation of $\mathbf{p}$ with equation 4 is very sensitive to the presence of outlier points: a high error value from one point will severely influence the values of $\mathbf{p}$. We use M-estimators [47] to solve this problem. This technique has been applied to ASM and AAM in previous works [48, 49], so it is only briefly presented here.

Let $\mathbf{s}$ be a shape, obtained by fitting each feature independently. The function to minimize is

$$arg \min_{\mathbf{p}} \sum_{i=1}^{2n} \rho(r^i, \theta) \tag{5}$$

where $\rho : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is an M-estimator, and θ is obtained from the standard deviation of the residues [50]. r^i is the residue for coordinate i of the shape

$$r^i = \mathbf{x}^i - (\mathbf{s}_o^i + \sum_{j=1}^{m} p_j \mathbf{s}_j^i)) \tag{6}$$

where $\mathbf{x}^i$ are the points of the shape $\mathbf{s}$, and $\mathbf{s}_j^i$ is the ith element of the vector $\mathbf{s}_j$.

Minimizing function 5 is a case of re-weighted least squared. The weight decreases more rapidly than the square of the residue, and thus a point with error tending to infinite will have zero weight in the estimation.

Several robust estimators have been tested: *Huber*, *Cauchy*, *Gaussian* and *Tukey* functions [50]. A study was made in [19] that resulted in similar performance for all of them in a similar scenario to that of this paper, and Huber function was chosen. Huber function performs correctly up to a number of outliers of 50% of the points.

We use the 20-point distribution of the BioID database [51]. Data from this database was used to train the model. This distribution places the points in some of the most salient locations of the face, and has been used in several other works [40].

4. Tests and results

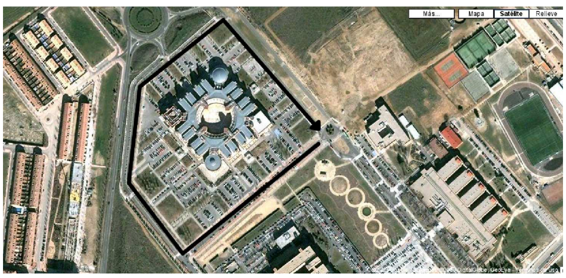
This section presents the video sequences used to test different tracking algorithms in a driving scenario. The dataset contains most actions that appear in everyday driving situations. A comparison between our approach and SMAT is presented. Additionally, we compare R-SMAT results with the recently introduced Stacked Trimmed ASM (STASM).

4.1. Test set

Driving scenarios present a series of challenges for a face tracking algorithm. Drivers move constantly, rotate their head (self-occlusing part of the face) or occlude their face with their hands (or other elements such as glasses). If other people are in the car, talking and gesturing are common. There are also constant background changes and, more importantly, frequent illumination changes, produced by shadows of trees or buildings, streets lights, other vehicles, etc. A considerable amount of test data is needed to properly evaluate the performance of a system under all these situations.

A new video dataset has been created, with sequences of subjects driving outdoor, and in a simulator. The RobeSafe Driver Monitoring Video (RS-DMV) dataset contains 10 sequences, 7 recorded outdoors (*Type A*) and 3 in a simulator (*Type B*).

Outdoor sequences were recorded on RobeSafe's vehicle moving at the campus of the University of Alcala. Drivers were fully awake, talked frequently with other passengers in the vehicle and were asked to look regularly to the rear-view mirrors and operate the car sound system. The cameras are placed over the dashboard, to avoid occlusions caused by the wheel. All subjects drove the same streets, shown in figure 2.



(a) Trayectory of the vehicle during recordings

Figure 2: Trayectory of the vehicle (map from *maps.google.com*)

The length of the track is around 1.1 km. The weather conditions during the recordings were mostly sunny, which made noticeable shadows appear on the face. Figure 3 shows a few samples from these video sequences.

Figure 3: Samples of outdoor videos

Type B sequences were recorded in a realistic truck simulator. Drivers were fully awake, and were presented with a demanding driving environment were many other vehicles were present and potentially dangerous situations took place. These situations increase the probability of small periods of distraction leading to crashes or near-crashes. The sequences try to capture both distracted behaviour and the reaction to dangerous driving situations.

A few images from *Type B* sequences can be seen in figure 4. The recording took place in a low-light scenario that approached nighttime conditions. This forced the camera to increase exposure time to a maximum, which lead to motion blur being present during head movements. Low power near-IR illumination was used in some of the sequences to increase the available light.

Figure 4: Samples of sequences in simulator

The outdoor sequences are around 2 minutes long, and sequences in the simulator are close to 10 minutes in length. The algorithms in this paper were tested on images of approximately 320×240 pixels, but high resolution images were acquired so they can be used in other research projects. The images are 960×480 pixels for the outdoor sequences and 1392×480 for the simulator sequences, and are stored without compression. Frame rate is 30 frames per second in both cases. The camera has a 2/3" sensor, and used 9mm standard lenses. Images are grayscale. The recording software controlled camera gain using values of the pixels that fell directly on the face of the driver.

The RS-DMV is publicly available, free of charge, for research purposes. Samples and information on how to obtain the database are available at the authors' webpage[1].

4.2. Performance evaluation

Performance of the algorithms is evaluated as the error between the estimated position of the features and their actual position, as given by a human operator. Hand-marking is a time consuming task, and thus not all frames in all videos have been marked. Approximately 1 in 30 frames (1 per second) has been marked in the sequences in RS-DMV. We call this frames *keyframes*.

We used the metric m_e, introduced by Cristinacce and Cootes [40]. Let $\mathbf{x}^i$ be the points of the ground-truth shape $\mathbf{s}$, and let $\hat{\mathbf{x}}^i$ be the points of the estimated shape $\hat{\mathbf{s}}$. Then,

$$m_e = \frac{1}{ns}\sum_{i=1}^{n} d^i, \qquad d^i = \sqrt{(\mathbf{x}^i - \hat{\mathbf{x}}^i)^T(\mathbf{x}^i - \hat{\mathbf{x}}^i)} \tag{7}$$

where n is the number of points and s is the inter-ocular distance. We also discard the point on the chin and the exterior of the eyes, because their location changes much from person to person. Moreover, the variance of their position when marked by human operators is greater than for the other points. Because only 17 points are used, we note the metric as m_e17. In the event of a tracking loss, of if the face can not be found, the value of m_e17 for that frame is set to ∞.

During head turns, the inter-eye distance reduces with the cosine of the angle. In these frames, s is not valid and is calculated from its value on previous frames.

Handmarked points and software used to ease the marking process are distributed with the RS-DMV dataset.

4.3. Results

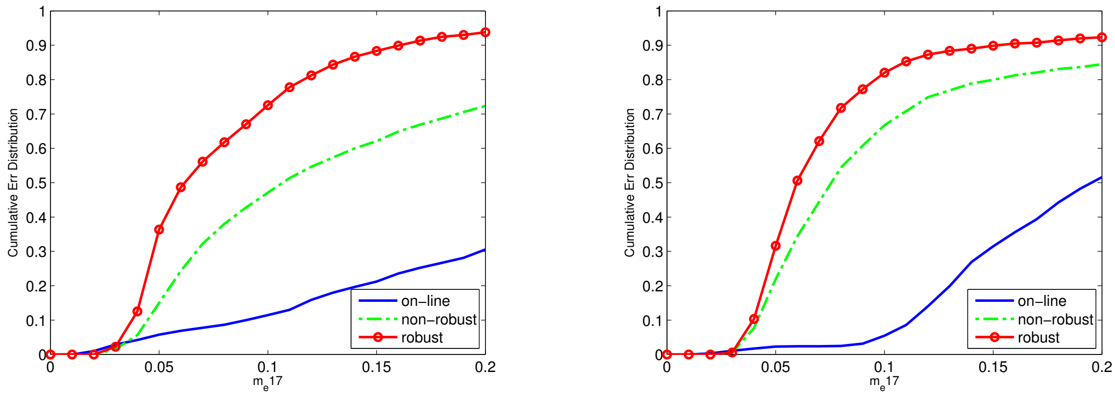
We tested the performance of R-SMAT approach on the RS-DMV dataset, as well as that of SMAT. We compared these results with those obtained by STASM, using the implementation in [2].

One of the most remarkable problems of (R-)SMAT is that it needs to be properly initialized, and the first frames of the sequence are key to building a good model. We propose STASM to initialize (R-)SMAT in the first frame. STASM has been shown to be very accurate when the face is frontal. Nonetheless, a slightly incorrect initialization will make (R-)SMAT track the (slightly) erroneous points. To decouple this error from the evaluation of accuracy of (R-)SMAT in the tests, the shape was initialized in the first frame with positions from the ground-truth data. At the end of this section, the performance of R-SMAT with automatic initialization is evaluated.

[1] www.robesafe.com/personal/jnuevo

First, a comparison of the shape models is presented. With the best shape model, the original clustering algorithm and the proposed alternative are evaluated. Results are presented for outdoor and simulator sequences separatedly, as each has specific characteristics on their own.

The incremental shape model of SMAT was found to produce much higher error than the pre-learned model. Figure 5 shows the cumulative distribution error of the incremental shape model (*on-line*) with the robust pre-learned model (using Huber function) (*robust*). For comparison purposes, the figure also shows the performance for the pre-learned shape model fitted using a L_2 norm (*non-robust*). All models use *leaderP* clustering, and patches of 15×15 pixels.



(a) *Type A* (b) *Type B*

Figure 5: Performance of different shape models, with *leaderP* clustering

Clear improvements in performance are made by the change to a pre-learned model with robust fitting. The robust, pre-learned shape model is very important in the first frames, because it allows the model to have bigger certainties that the patches that are being included correspond to correct positions. Robust shape model is used in the rest of the experiments in this paper. Figure 6 shows the plot of the m_e17 distance of both models in a sequence. A clear example of the benefits of the robust model is depicted in figure 7. The online model diverges as soon as a few points are occluded by the hand, while the robust model keeps track of the face. The method is also able to keep track of the face during head rotations, although with increased fitting error. This is quite remarkable for a model that has only been trained with fully frontal faces.

Figure 8 shows the performance of the original SMAT clustering compared with the proposed *leaderP* clustering algorithm, as implemented in R-SMAT.

R-SMAT presents much better performance than the original SMAT clustering. This is specially clear in 8(b). We stated in 3.1 that the original clustering method could lead to overfitting, and *type B* sequences are specially prone to this: patches are usually dark and do not change much from frame to frame, and the subject does not move frequently. When a movement takes place, it leads to high error values, because the model has problems finding the features.

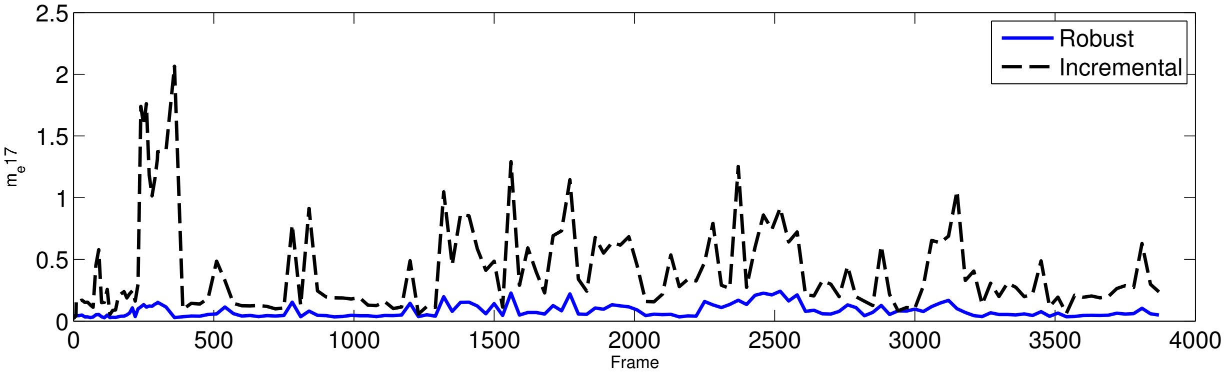
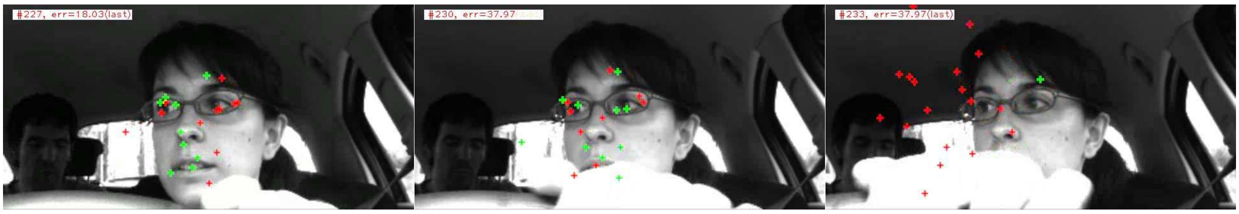
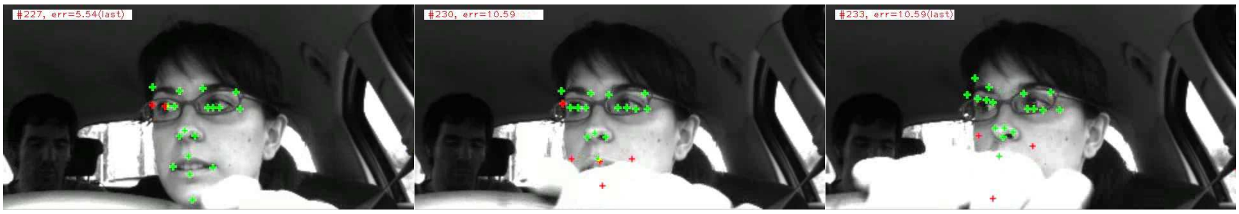


Figure 6: m_e17 error for a sequence



(a) Frame 227 - online (b) Frame 230 - online (c) Frame 233 - online



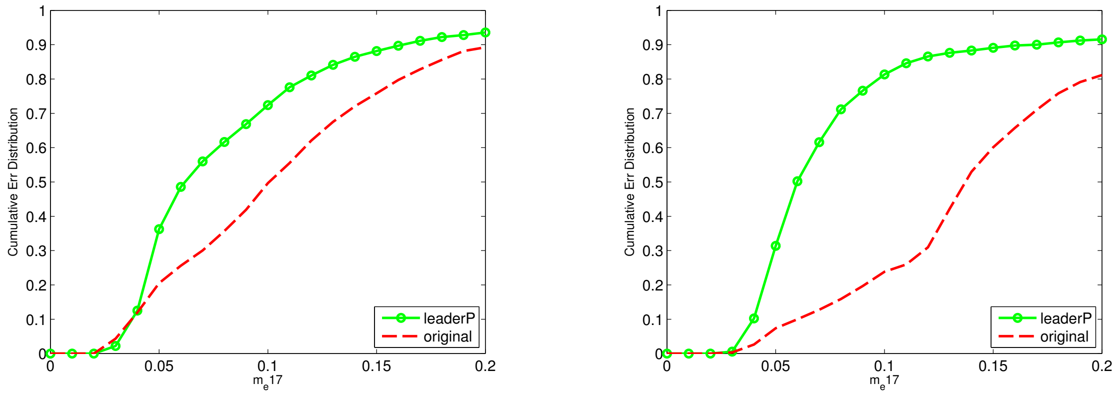
(d) Frame 227 - robust (e) Frame 230 - robust (f) Frame 233 - robust

Figure 7: Samples of *type A* sequence #1. Outlier points are drawn in red.

Table 1 shows the tracking losses of SMAT and R-SMAT, as a percentage of the *keyframes* in the sequences. Tracking losses were monitored by counting the points inside the face area, detected with Viola&Jones algorithm [52]. Tracking was considered lost when more than 33% of the points were out of the box, or when the rotation of the model exceeded a pre-set value. The model was then repositioned, simply by centering it on the Viola&Jones box.

		Mean	Maximum	Minimum
R-SMAT	*Type A*	0.99%	2.08%(seq. #7)	0%(seq. #1,#2,#6)
	Type B	0.71%	1.96%(seq. #9)	0%(seq. #10)
SMAT	*Type A*	1.77%	5.03%(seq. #4)	0%(seq. #1,#2,#5)
	Type B	1.03%	2.45%(seq. #9)	0%(seq. #10)

Table 1: Track losses for different clustering methods



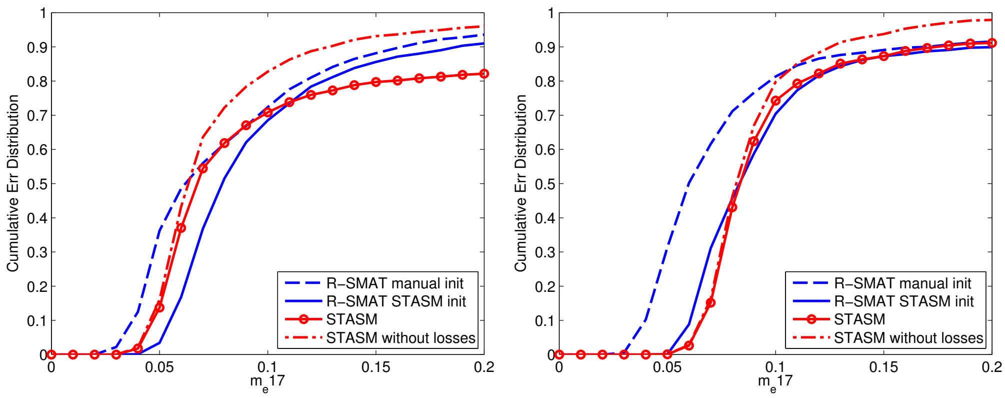
(a) *Type A* (b) *Type B*

Figure 8: Comparison of the performance of clustering algorithms

4.3.1. R-SMAT with automatic initialization

Results presented above have been obtained initializing SMAT and R-SMAT with landmarks from the handmarked ground-truth data. In a real scenario, an automatic algorithm would be used to initialize SMAT and R-SMAT.

We have used STASM for this task. STASM has demonstrated high accuracy, but only works properly when the face is frontal to the camera, and does not find the face otherwise. Another problem, critical to our application, is that it does not work in real time. However, a one-time delay can be considered acceptable. STASM was run on the first frame of each sequence, and its estimation seeded the position of R-SMAT in that video. Figure 9 plots the error distributions of R-SMAT when initialized with STASM and from ground-truth (manual). The error of STASM is also shown.



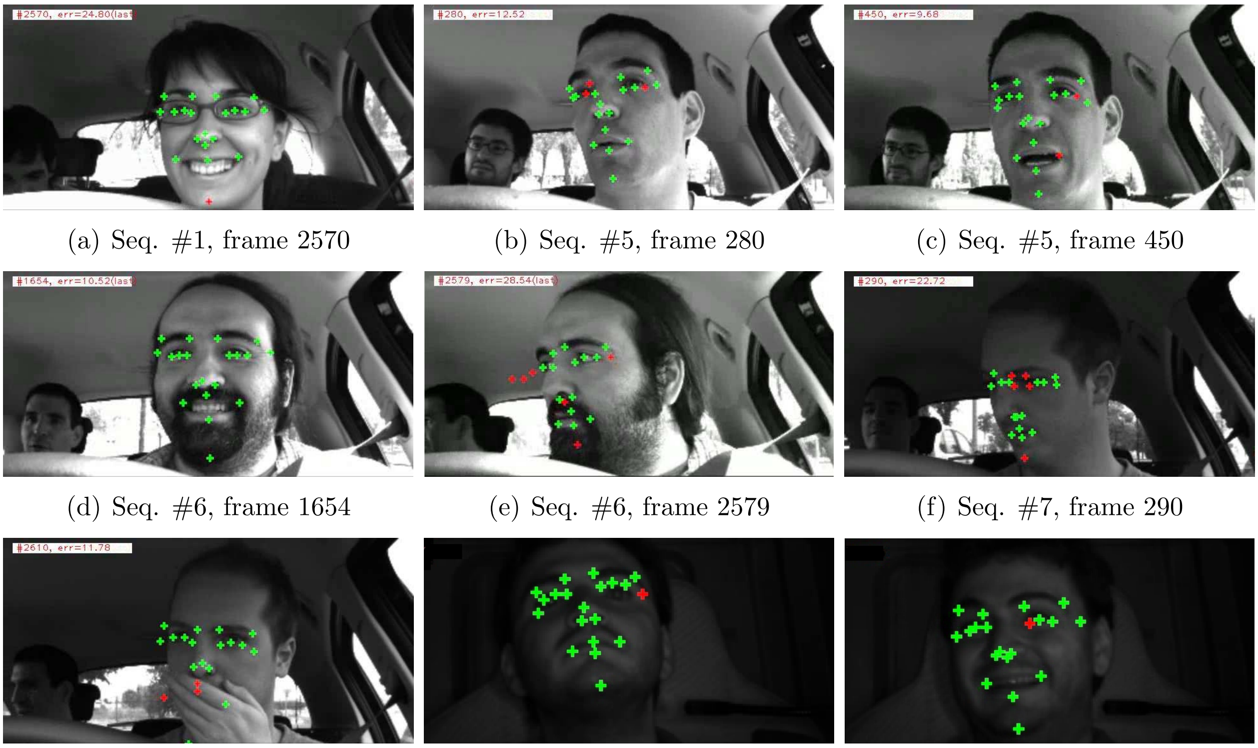
(a) *Type A* (b) *Type B*

Figure 9: Comparison of the performance of STASM and SMAT

As expected, the figure shows the results worsen for both types of sequences. But the lost accuracy is relatively small, with a 5% loss at $m_e17 = 0.1$ for *type A* sequences, and

10% loss for *type B*. The mean of the m_e17 error of STASM in the first frame is 0.0571 for *type A* sequences and 0.0805 for *type B* .

STASM is plotted in figure 9 with and without considering frames where the face was not found (losses). For all types of sequences, R-SMAT initialized manually outperforms STASM when losses are considered. Expectedly, STASM shows better accuracy than R-SMAT when tracking is not lost (i.e., when the face is frontal). R-SMAT initialized with STASM performs almost identically as STASM in figure 9(b), and slightly worse for *type A* sequences.



(a) Seq. #1, frame 2570 (b) Seq. #5, frame 280 (c) Seq. #5, frame 450

(d) Seq. #6, frame 1654 (e) Seq. #6, frame 2579 (f) Seq. #7, frame 290

(g) Seq. #7, frame 2610 (h) Seq. #9, frame 7440 (i) Seq. #9, frame 11730

Figure 10: Examples from sequences with R-SMAT fitted

Figure 10 shows a few frames with R-SMAT fitted to the face of the drivers in moments of the sequences that reflect some of the challenges the system has to face. Figures 10(a)-10(d) contain drivers talking and gesturing. Drivers in sequences #5 and #6 talk frequently and no tracking loss results from these actions. The system is able to work with drivers that also wear a beard, as in figure 10(d). Examples of head turns appear in figures 10(e) and 10(f). Most occlusions in the sequences are caused by the presence of a hand in front of the camera. If the occlusion is partial as in figure 10(g), the tracker is able to correctly position the model.

Samples of R-SMAT fitted to one of the simulator sequences appear in figures 10(h) and 10(i). Despite the low-light conditions, a low fitting error is obtained. The driver in the sequences talks and gestures frequently.

Figures 11 and 12 plot the error for R-SMAT and STASM for sequences #6 and #7. Dots on the STASM curve mark keyframes where the face was not found. A case of quick illumination change, due to shadows of trees by the road is found in sequence #6 around frame 1400. A R-SMAT track loss can be observed around frame 2400 of the latter sequence, due to a total occlusion of the face. Figure 12 also shows that fitting error is higher during head turns, but in most cases track is not lost.

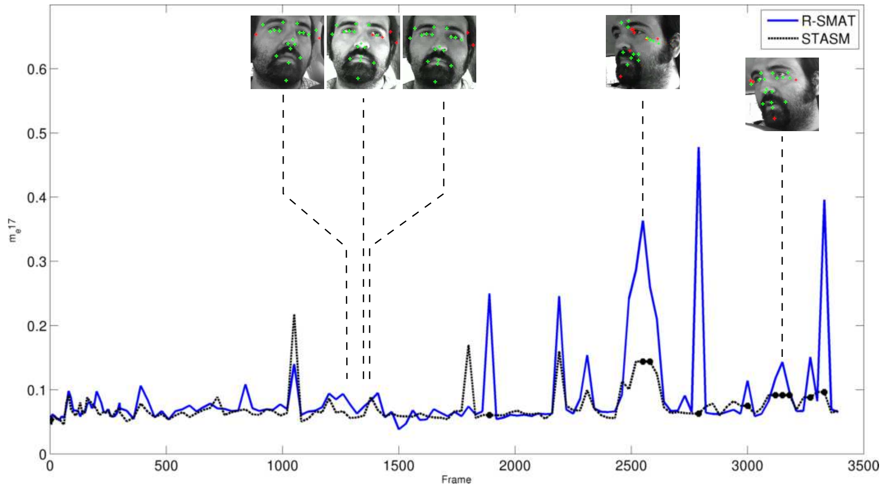


Figure 11: Error plots for STASM and R-SMAT in sequence #6

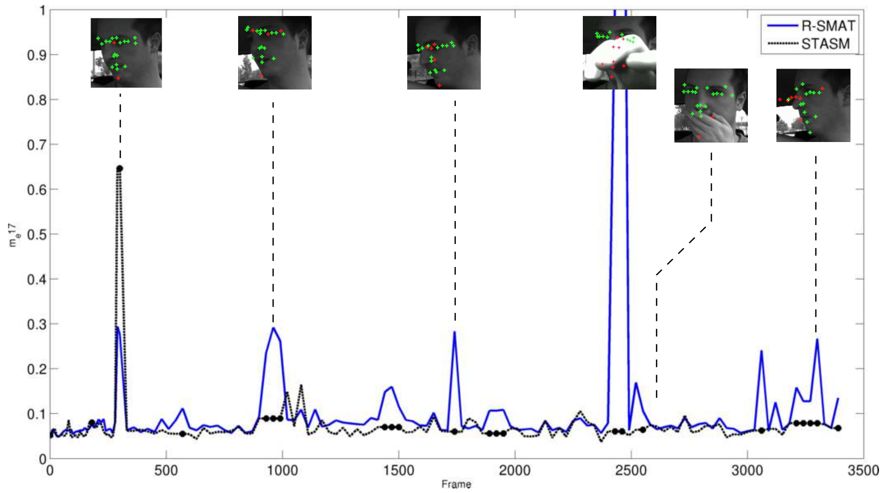


Figure 12: Error plots for STASM and R-SMAT in sequence # 7

4.4. Timings

One of the most important requirements for R-SMAT is for it to run in real-time. Table 2 summarizes the average execution speed for R-SMAT in frames per second, for some representative configurations. The worst frame processing times are close to the limit, but these are extreme cases that occur infrequently. Processing times for STASM are also included for comparison. STASM executes the whole initialization process for each frame, and does not use the position found in the previous frame. This would result in a shorter search time.

Configuration	Mean (fps)	Sdv (fps)	Worst frame (fps)
R-SMAT	112.56	32.80	36.86
STASM	2.17	0.13	1.96

Table 2: Execution time of R-SMAT and STASM in frames per second

The tests were run on a Xeon 2.2 GHz, running GNU/Linux, with GCC 4.2 as compiler. Multi-threading was not used and compiler optimizations were disabled (`-O0`). Times on the table refer to the actual tracking and the tracking loss detection, and do not consider time employed in the display of results, loading of data and saving results to the hard drive.

5. Conclusions and future work

This paper has presented a face tracking method based on automatic appearance modeling, to be used as part of a driver monitoring application.

Monitoring a driver with computer vision is a complex task, and proper performance evaluation of a method requires a comprehensive set of test data. A new video dataset (RS-DMV) has been created, comprised of sequences recorded in a real scenario, and in a truck simulator in low light conditions. In the first set of sequences, subjects were asked to drive a vehicle at the University campus. Drivers in the simulator were fully awake, and were presented with dangerous situations that would highlight distractions. RS-DMV dataset has been used to test the methods in this paper, and is freely available for research purposes.

The proposed face tracking method is an active model. The shape model is built offline from handmarked data, and Huber fuction is used for fitting. The texture model is created online using incremental clustering, in a similar fashion to SMAT. We have presented and alternative incremental clustering algorithm, which addresses some of the weaknesses of the SMAT proposal. The improvements of R-SMAT over the original SMAT have been evaluated, and the performance of R-SMAT and STASM has been compared on the sequences in RS-DMV. R-SMAT is able to process more than 100 frames per second, and obtains similar accuracy to STASM .The source code of R-SMAT is available from the authors.

Future work will explore ways to make R-SMAT fully autonomous by improving the incremental shape model. Texture clustering has shown to be reliable, but better techniques to remove outliers from the model are needed. Including a multi-scale approach to appearance modeling and model fitting would be of help in other scenarios where the size of the face changes noticiably. The R-SMAT has been used to track and model faces in this paper, but can be extended to other deformable objects. The RS-DMV dataset will be extended with more sequences, more drivers and more diverse scenarios. Finally, the R-SMAT is to be made part of a driver monitoring system.

Acknowledgments

The authors would like to thank Pedro Jiménez, Ivan García and Noelia Hernández of RobeSafe for their work in recording the sequences, Sebastian Bronte for his help in marking the images, as well as the drivers that participated. The outdoor recordings were made under project MOVI^{2}CON (`TRA2005-08529-C02-02`) and the simulator recordings under CABINTEC project (`PSE-370100-2007-2`). This work was supported in part by the Spanish Ministry of Science and Innovation under DRIVER-ALERT Project (`TRA2008-03600`), and Comunidad de Madrid under project RoboCity2030 (`S-0505/CPI/000176`). J. Nuevo was working under a researcher training grant from the Education Department of the Comunidad de Madrid and the European Social Fund.

[1] R. Dowson, N.D.H.; Bowden, Simultaneous modeling and tracking (SMAT) of feature sets, in: IEEE Conference on Computer Vision and Pattern Recognition 2005, Vol. 2, 2005, pp. 99–105.

[2] S. Milborrow, F. Nicolls, Locating facial features with an extended active shape model, in: ECCV, 2008, pp. 504–513, `http://www.milbo.users.sonic.net/stasm`.

[3] T. A. Dingus, S. Klauer, V. L. Neale, A. Petersen, S. E. Lee, J. Sudweeks, M. A. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z. R. Doerzaph, J. Jermeland, R. Knipling, The 100-car naturalistic driving study, Tech. rep., Virginia Tech Transportation Institute, NHTSA (Apr. 2006).

[4] A. Kircher, M. Uddman, J. Sandin, Vehicle control and drowsiness, Tech. Rep. VTI-922A, Swedish National Road and Transport Research Institute (2002).

[5] Volvo Car Corp., Driver alert control, `http://www.volvocars.com` (2008).

[6] DaimerAG, Attention assist, http://www.daimler.com (Jun. 2009).

[7] H. Ueno, M. Kaneda, M. Tsukino, Development of drowsiness detection system, in: Proceedings of Vehicle Navigation and Information Systems Conference, 1994, pp. 15–20.

[8] E. Murphy-Chutorian, M. Trivedi, Head pose estimation in computer vision: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 31 (4) (2009) 607–626. doi:10.1109/TPAMI.2008.106.

[9] P. Smith, M. Shah, N. da Vitoria Lobo, Determining driver visual attention with one camera, IEEE Trans. Intell. Transp. Syst. 4 (4) (2003) 205–218.

[10] J. Wu, T. Chen, Development of a drowsiness warning system based on the fuzzy logic images analysis, Expert Systems with Applications 34 (2) (2008) 1556–1561.

[11] Y. Matsumoto, A. Zelinsky, An algorithm for real-time stereo vision implementation of head pose and gaze direction measurements, in: Procs. IEEE 4th Int. Conf. Face and Gesture Recognition, 2000, pp. 499–505.

[12] T. Victor, O. Blomberg, A. Zelinsky, Automating the measurement of driver visual behaviours using passive stereo vision, in: Procs. Int. Conf. Series Vision in Vehicles VIV9, Brisbane, Australia, 2001.

[13] M. Kutila, Methods for Machine Vision Based Driver Monitoring Applications, Ph.D. thesis, VTT Technical Research Centre of Finland (2006).

[14] T. D'Orazio, M. Leo, C. Guaragnella, A. Distante, A visual approach for driver inattention detection, Pattern Recognition 40 (8) (2007) 2341–2355.

[15] Seeing Machines, Driver state sensor, http://www.seeingmachines.com/dss.html (Aug. 2007).

[16] SmartEyeAG, AntiSleep, `www.smarteye.se` (2009).

[17] Smart Eye AB, Image capturing device with reflex reduction, Patent, EP 1349487B1 (Dec. 2001).

[18] Q. Ji, Z. Zhu, P. Lan, Real-time nonintrusive monitoring and prediction of driver fatigue, IEEE Trans. Veh. Technol. 53 (4).

[19] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, E. López, Real-time system for monitoring driver vigilance, IEEE Trans. Intell. Transp. Syst. 7 (1) (2006) 1524–1538.

[20] M. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 24 (1) (2002) 34–58. doi:http://doi.ieeecomputersociety.org/10.1109/34.982883.

[21] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific

linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720, special Issue on Face Recognition.

[22] J. M. Buenaposada, E. Muñoz, L. Baumela, Recognising facial expressions in video sequences, Pattern Analysis and Applications 11 (1) (2008) 101–116.

[23] G. D. Hager, P. N. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Trans. Pattern Anal. Mach. Intell. 20 (10) (1998) 1025–1039. doi:http://dx.doi.org/10.1109/34.722606.

[24] J. M. Buenaposada, E. Muñoz, L. Baumela, Efficiently estimating facial expression and illumination in appearance-based tracking, in: Proc. British Machine Vision Conference, Vol. 1, 2006, pp. 57–66.

[25] F. Jurie, M. Dhome, Hyperplane approximation for template matching, IEEE Trans. Pattern Anal. Mach. Intell. (2002) 996–1000.

[26] S. Baker, I. Matthews, Equivalence and efficiency of image alignment algorithms, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 2001, pp. 1090–1097.

[27] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active Shape Models-Their Training and Application, Computer Vision and Image Understanding 61 (1) (1995) 38–59.

[28] I. Dryden, K. Mardia, Statistical Shape Analysis, John Wiley & Sons, 1998.

[29] T. F. Cootes, G. J. Edwards, C. J. Taylor, Active appearance models, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 681–685.

[30] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: International Joint Conference on Artificial Intelligence, Vol. 3, 1981, pp. 674–679.

[31] A. Jepson, D. Fleet, T. El-Maraghi, Robust Online Appearance Models for Visual Tracking, IEEE Trans. Pattern Anal. Mach. Intell. (2003) 1296–1311.

[32] Z. Yin, R. Collins, On-the-fly Object Modeling while Tracking, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, 2007, pp. 1–8.

[33] R. Collins, Y. Liu, M. Leordeanu, Online Selection of Discriminative Tracking Features, IEEE Trans. Pattern Anal. Mach. Intell. (2005) 1631–1643.

[34] S. Avidan, Ensemble Tracking, IEEE Trans. Pattern Anal. Mach. Intell. (2007) 261–271.

[35] H. Grabner, H. Bischof, On-line boosting and vision, in: Proc. CVPR, Vol. 1, 2006, pp. 260–267.

[36] M. Pham, T. Cham, Online Learning Asymmetric Boosted Classifiers for Object Detection, in: Computer Vision and Pattern Recognition CVPR, 2007, pp. 1–8.

[37] J. Pilet, V. Lepetit, P. Fua, Real-time non-rigid surface detection, in: IEEE Conference on Computer Vision and Pattern Recognition 2007, San Diego, CA, 2005, pp. 822–828.

[38] I. Matthews, T. Ishikawa, S. Baker, The Template Update Problem, IEEE Trans. Pattern Anal. Mach. Intell. (2004) 810–815.

[39] S. Segvic, A. Remazeilles, F. Chaumette, Enhancing the point feature tracker by adaptive modelling of the feature support, in: European Conf. on Computer Vision, ECCV'2006, Vol. 3952 of Lecture Notes in Computer Science, Graz, Austria, 2006, pp. 112–124.

[40] D. Cristinacce, T. Cootes, Feature Detection and Tracking with Constrained Local Models, in: 17th British Machine Vision Conference, 2006, pp. 929–938.

[41] L. Bergasa, J. Buenaposada, J. Nuevo, P. Jimenez, L. Baumela, Analysing Driver's Attention Level using Computer Vision, in: Intelligent Transportation Systems. ITSC 2008. 11th International IEEE Conference on, 2008, pp. 1149–1154.

[42] N. Dowson, R. Bowden, N-tier simultaneous modelling and tracking for arbitrary warps, in: Proc. of the 17th British Machine Vision Conference. British Machine Vision Association, Vol. 1, 2006, p. 6.

[43] J. Hartigan, Clustering Algorithms, John Wiley & Sons, Inc. New York, NY, USA, 1975.

[44] H. Spath, V. Bull, Cluster analysis algorithms for data reduction and classification of objects, Ellis Horwood, 1980.

[45] R. Gross, I. Matthews, S. Baker, Generic vs. person specific active appearance models, Image and Vision Computing 23 (11) (2005) 1080–1093.

[46] J. Gower, Generalized procrustes analysis, Psychometrika 40 (1) (1975) 33–51.

[47] P. J. Huber, Robust Statistics, Wiley Series in Probability and Mathematical Statistics, Wiley-

Interscience, 1981.
[48] M. Rogers, J. Graham, Robust active shape model search, Lecture Notes in Computer Science (2002) 517–530.
[49] R. Gross, I. Matthews, S. Baker, Constructing and fitting active appearance models with occlusion, in: Proceedings of the IEEE Workshop on Face Processing in Video, 2004, p. 72.
[50] Z. Zhang, Parameter estimation techniques: A tutorial with application to conic fitting, Image and vision Computing 15 (1) (1997) 59–76.
[51] O. Jesorsky, K. Kirchberg, R. Frischholz, et al., Robust face detection using the hausdorff distance, Proceedings of Audio and Video based Person Authentication (2001) 90–95http://www.bioid.com/downloads/facedb/.
[52] P. Viola, M. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.