Jiménez, P., Nuevo, J., Bergasa, L.M. & Sotelo, M.A. 2009, "Face tracking and pose estimation with automatic three-dimensional model construction", IET Computer Vision, vol. 3, no. 2, pp. 93-102

Available at http://dx.doi.org/10.1049/iet-cvi.2008.0057

Face Tracking and Pose Estimation with Automatic 3D Model Construction

Pedro Jiménez, Jesús Nuevo, Luis M. Bergasa
Department of Electronics
University of Alcala
CAMPUS. 28805 Alcalá de Henares (Madrid), Spain.
E-Mail: pjimenez,jnuevo,bergasa@depeca.uah.es

***Abstract*—This paper presents a method for robustly tracking and estimating the face pose of a person using stereo vision. The method is invariant to identity and does not require previous training. A face model is automatically initialised and constructed on-line: a fixed point distribution is superposed over the face when it is frontal to the cameras, and several appropriate points close to those locations are chosen for tracking. Using the stereo correspondence of the cameras, the 3D coordinates of these points are extracted, and the 3D model is created. The 2D projections of the model points are tracked separately on the left and right images using SMAT. RANSAC and POSIT are used for 3D pose estimation. Head rotations up to $\pm 45^{\circ}$ are correctly estimated. The approach runs in real time. The purpose of this method is to serve as the basis of a driver monitoring system, and has been tested on sequences recorded in a moving car.**

I. Introduction

Face detection and tracking is a very active research field in computer vision, and a comprehensive number of methods have been developed [1]. Face detection is also the first step in many other algorithms in face recognition, tracing, expression analysis and other areas of computer vision [2], [3], [4]. Face pose estimation has attracted interest for its usefulness in different applications. It is an important cue of where the person is directing his or her attention, and thus has been widely used in Human-Machine Interface applications, sometimes coupled with gaze estimation [5]. It is a principal component of many driver inattention monitoring systems [6], [7], [8].

Determining face pose is a complex problem, and numerous methods have been proposed to estimate it [9], [10], [11], [12]. Obtaining the pose from 2D images is difficult because many factors appear coupled. Face movement is subject not only to rigid variations (pose), but also non-rigid deformations (expressions). These deformations can be separated in 3D, but doing so in 2D is a challenging problem, because they are not linearly separable [13], [14]. Several works, such as the one presented in this paper, are nonetheless able to obtain positive results assuming a rigid face [15]. In 2D images, self-occlusions also have to be indirectly calculated and robustly handled.

Most approaches to the problem of face pose estimation use a face model of various kinds. Among the most common are 2D appearance models [16], 2D+3D [14], 3D models from range images [11], and patch-based models [17]. Estimating the pose with a model is usually a step in the process of fitting the model to the image or range scan. These models are created with an offline training process that involves large amounts of samples and it can be very time-consuming.

Our approach is similar to [17] in that we use a set of 3D points with a patch associated to model the appearance of the face around each point. Our model is, however, a rigid model that does not consider face deformations due to expressions. This is a valid assumption when the magnitude of the expressions is small and if enough points that deform very slightly on expressions are taken, i.e., eye corners or points over the nose. We propose a stereo camera system that can track different users without prior training. The model is automatically initialised on the first frame of the video sequence, locating adequate features for tracking over the face of the subject. The 3D coordinates of these points are calculated using the stereo correspondence of the cameras. To reduce the computational load, points are tracked semi-independently on each image with a modified SMAT [18], and the pose recovered with POSIT [19]. Points incorrectly tracked are removed from the estimation using RANSAC [20]. The system is able to track a driver's face robustly in real conditions. Experimen-

tal results and an analysis of the performance are presented. This paper extends the work presented in [21].

II. Related Works

Many different approaches have been made to the face pose estimation problem. A comprehensive survey of the state of the art has been recently carried out by Murphy-Chutorian and Trivedi in [22]. In this section we present some representative works, and refer to [22] for more information. One classical approach to this problem was presented by La Cascia *et al.* in [23]. Their approach uses a 3D cylinder model and optical flow to estimate the head pose. The system was tested on a few indoor sequences with controlled head turns. In recent years, Active Appearance Models [16] have been used to estimate the object pose, in 2D [24] and 2D+3D spaces [14]. Appearance models represent the face as a flexible object, with changes in position, shape and appearance modelled as a linear subspace of all possible variations. They have been shown to work reliably in many different scenarios. Although efficient versions have been introduced [14], their fitting algorithms are computationally expensive, and some exhibit fitting convergence problems when the face shape or illumination change rapidly. Three dimensional face models [25] include pose estimation as part of the fitting process.

In [26], Morency *et al.* use depth and intensity view-based eigenspaces to build a prior model from the first frame that is then robustly tracked. Murphy-Chutorian and Trivedi [6] presented a system based on Localized Gradient Orientation histograms integration with Support Vector Machines for regression that has obtained good results in tests performed with drivers in a moving car.

Several methods have been presented that work on range scan data [11]. 3D acquisition systems provide accurate and dense data, but the vast amount of data requires powerful parallel processors (GPU), and not always can be processed in real-time [25].

Instead of working with dense data, in [27] detection of non-rigid surfaces is done based on keypoint recognition. This algorithm works in real time, and its keypoint classifier can be trained within minutes [28].

Considering the face a rigid object simplifies its modelling and tracking. The reflection of near-infrared light on the subject's eyes (*red-eye effect*) has been used in several works [12], [29]. This technique has also been used in [29] to estimate the pose of the face. These works obtained accurate tracking and estimations in real time in indoor tests. However, the *red-eye effect* may not be visible outdoors where sunlight is present. Also, continuous exposure to near-IR lighting is a known cause of eye fatigue, so users may not be able to use these systems for extended periods of time.

III. Face model and pose estimation

Automatically creating the face model on the first frame of the sequence removes the offline training process. It also allows to work with an specific model for the subject in the sequence. On the downside, not having a trained model makes identifying the different parts very difficult, although that is outside the scope of this work.

Our model is formed by a set of up to 30 3D points of the face. These points present adequate characteristics for tracking and are found with Harris detector [30]. The patches around the 2D projections of these points on each camera are tracked on each frame, using the Simultaneous Modelling and Tracking (SMAT) algorithm. 3D pose is obtained from the 2D points using POSIT, redundantly for both cameras to improve robustness.

Tracking may fail for some points on each frame. RANSAC is used to reject erroneous points from the estimation of the pose. After a set of correctly tracked points (inliers) is obtained, the position of the outlier points is set accordingly to the estimated pose. A diagram of the whole process is shown in Fig. 1.

A. 3D Face model creation

The camera model is referenced to a coordinate system affixed to the right camera. Within this system, the face pose is defined as a translation vector and a pointing vector, normal to the face. On model initialisation, it is set to $\vec{v}_{ini} = (x, y, z) = (0, 0, -1)$, as shown in Fig. 2. The translation vector points to the centre of the model.

The model points are referenced to another coordinate system, with origin on the central point of the model. $\vec{X}$ axis is the horizontal axis, and grows to the right of the image. $\vec{Y}$ axis is vertical, and grows down the frame, and the $\vec{Z}$ axis is perpendicular to the image plane and grows to the rear of the scene, so the nose of the driver should have the most negative z value.

Model creation begins with a face localisation step. The Viola & Jones [31] algorithm is used on both images to localise the position of a frontal face within

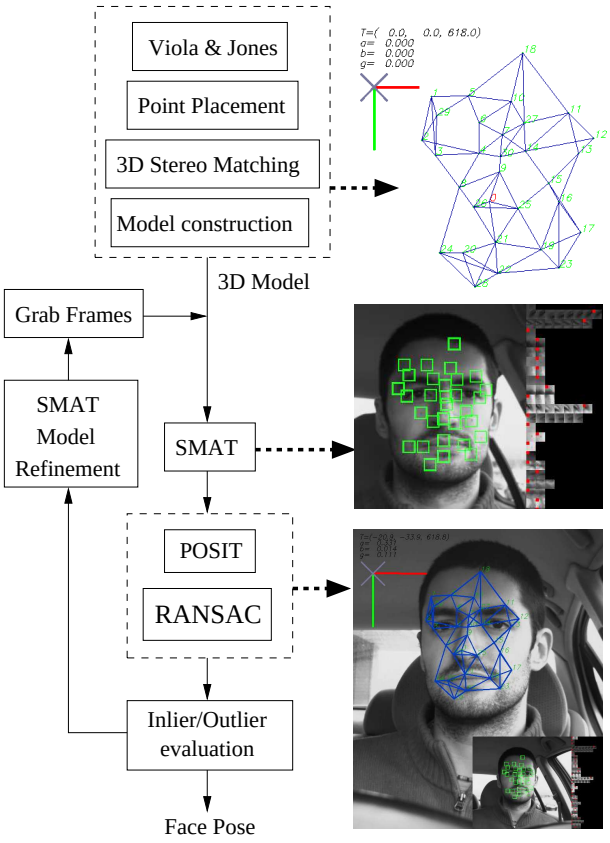


Fig. 1. System block diagram

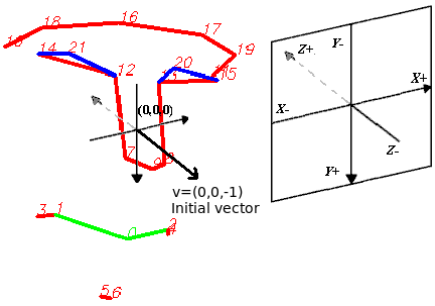


Fig. 2. Face Model, coordinate system and initial model vector

the camera frames. This algorithm returns a box that encloses the face. We reduce the size of this box by a factor experimentally obtained, so that it encloses the face with bigger certainty, as shown in Fig. 3.

The algorithm initialisation requires the subject to have a frontal pose to the cameras for a few frames at the beginning of system operation. At this moment, the model is considered to have a pointing vector $\vec{v}_{ini} = (x, y, z) = (0, 0, -1)$. If the user is not correctly positioned, the difference between the real pose vector at the initialisation step and $\vec{v}_{ini}$ will appear as a constant offset error.

The face model is defined by up to thirty points that are tracked over successive frames. To choose appropriate points, a predefined standard face pattern is scaled and placed over the detected inner box containing the face on the left camera image. These points may not fall over any good feature to be used for tracking on the specific user's face, so Harris algorithm [30] is used to locate features with good contrast and tracking characteristics in the vicinity of each pattern point.. Stereo correspondence of these points over the other camera are obtained applying epipolar restrictions, and are used to calculate its 3D coordinates. The stereo camera system was calibrated using the Camera Calibration Toolbox for MATLAB.

The predefined pattern that is placed over the face is similar to that shown in Fig. 2, and has most of its points around the eye area. An example of the resulting model can be seen in Fig. 1. This feature point location scheme provides great flexibility to the model, but it is very dependant on the success of the Harris algorithm in finding valid points. For users with few distinct points on their face, selected points tend to concentrate around the eyes and mouth. On the other hand, users with facial hair or other features such as moles will have the point set spread over their face more evenly. Both cases have potential problems associated. In the former, estimating the pose from points that are close together is more difficult when tracking inaccuracies appear, because the signal-to-noise ratio is lower. In the latter case, facial hair may be very similar across a significant part of the face, making tracking failures more probable.

The model is built with the 3D coordinates of the feature points. The model origin is then moved to the closest point to the centre of mass of the model, so that the initial 3D coordinates of the points are independent of the initial face position and distance to the camera, and the initial pose vector is set to $\vec{v}_{ini}$.

B. Model self-occlusion

The face model is subject to self-occlusion when the head exceeds a certain range. Some model points may not be visible, or appear too distorted to be correctly tracked. To detect such points, a hidden-point pattern is created after the model initialisation. Each point is associated a limit rotation angle within which the point is considered to be visible. When the face rotation angle is over the limit angle of a point, it is considered to be hidden and the point is not processed for tracking and pose estimation.

To create the hidden-point pattern, a circumference is adjusted to the (x, z) coordinates of each

(a) Left image (b) Right image

Fig. 3. Model Construction

model point, as shown in Fig. 4. The circumference is adjusted to minimise the function

$$w_k = \sum(\sqrt{(x_o - x_i)^2 + (z_o - z_i)^2} - R^2), \quad i = 1..30 \tag{1}$$

where (x_i, z_i) are the x and z coordinates of each model point, and (x_o, z_o) and R are the centre in the (x, z) plane and the radius of the circumference.

Each point of the model is considered to be hidden when its angle with respect to v_{ini} excesses ± 60 degrees. This is a simplification over calculating the visibility of each point. Setting a fixed threshold is however advisable considering that the model points are chosen automatically and may not represent face elements (such as the nose) that occlude other parts of the face when turns take place.

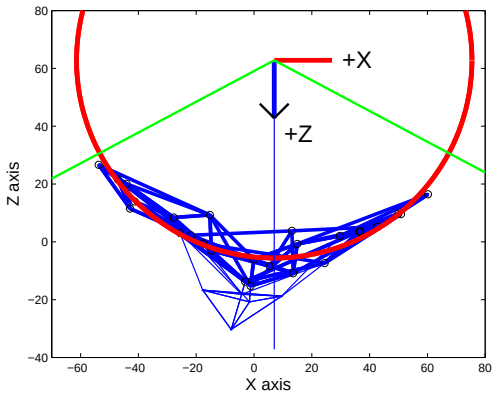


Fig. 4. Circumference fit to the face to get the limit angles

C. Tracking using SMAT

The Simultaneous Modelling and Tracking (SMAT) [18] is a recently developed technique for tracking objects in sequences. It is closely related to other techniques such as Constrained Local Models (CLM) [32], but it does not require any previous training. We briefly outline its main characteristics here, and some modifications we have included over the original work proposed by Dowson *et al.*

SMAT works by building a library of exemplars obtained from previous frames in the sequence. The exemplars in the library, image patches in our case, are clustered based on their relative distance, and the medians of the clusters are used for fitting the model to the next frame. A new exemplar is included in one of the clusters depending on the distance to their medians, or a new cluster is created if the new exemplar is too far away from the existing ones. As a group of similar exemplars, each of these clusters will approximately represent different appearances of the same feature of the object. The resulting mixture model is fitted to the next frame. Two tracking examples with a simple 4 patch model are shown in Fig. 5.

The SMAT algorithm is very flexible and can be used to track any kind of object. However, it has a few weaknesses. As it has no prior information, the model is defined on the first frames only by a few exemplars and may fail if occlusions or fast movements take place. In our method this problem is minimised because outliers are rejected and exemplars that incorrectly added to the model are removed. On an implementation level, the mixture model in SMAT has a bigger memory footprint that

equivalent models from other methods. Very similar patches can be redundantly kept in a cluster, as no compression or dimensionality reduction technique (such as Principal Component Analysis) is used.

In the original paper, Dowson *et al.* [18] included a model of structure for the distribution of the exemplars on the images, that is also built on-line. Nevertheless, our implementation only builds an appearance model, and the point distribution of the shape is constrained by the 3D model. Restricting learning to the appearance model reduces complexity and uncertainty, and allows to use robust methods, and discard points and exemplars that do not fit well, improving the overall robustness of the tracking. Further details are given in the section below.

The formulation of the SMAT algorithm is independent of distance measure used to compare the cluster medians with the incoming patches. A minimisation method is used to obtain the best matching position without evaluating the distances at every point in the vicinity of the position. We have used Zero mean Normalised Cross-Correlation (ZNCC) and Sum of Squared Differences (SSD) as distances, and Gauss-Newton and the Nelder-Mead simplex method [33] for the minimisation process. Of the two distance measures tested, the best performing was ZNCC. This distance is more robust to changes of illumination if those take place over the whole patch, which happens in most situations as the patches are small. All results presented in this paper were obtained using this distance measure.

D. Pose estimation

Given the new updated position of the 2D points for both left and right images, the 3D face pose is estimated from these 2D projections. However, the matching process may not succeed for all points, and can result in errors or drifting for some of them. These errors degrade the accuracy of the estimated pose. Thus, a robust method is required to estimate the best matching 3D face pose, so that points incorrectly tracked can be detected as outliers and safely discarded. We also consider that points that have been correctly tracked may have some random noise. The RANSAC algorithm is used to eliminate the outliers. 3D pose is obtained using DeMenthon's four point iterative pose estimation algorithm (POSIT) [19].

In each RANSAC iteration, seven points are randomly selected from the model, and used to calculate the pose (rotation matrix $\vec{R}$ and translation matrix $\vec{T}$) using the POSIT algorithm. The value of the pose is referenced to the pose in the first frame of the sequence. With this $\vec{R}$ and $\vec{T}$, all 3D original points of the model are projected over the image plane, and the Euclidean distance from the tracking point to the corresponding projected point is calculated. If this distance is less than a threshold, this point is considered to be correct, and marked as an inlier. The RANSAC algorithm runs for enough iterations to guarantee a 99% of success with 50% of outliers.

This process is performed over the left and right frames independently, and the final pose estimation is calculated from the pose estimations as a weighted sum, according to the expressions:

$$\vec{R}_{model} = \frac{\vec{R}_{right} \cdot I_l}{I_l + I_r} + \frac{\vec{R}^r_{left} \cdot I_r}{I_l + I_r}, \quad if\ I_r, I_l > I_{min} \tag{2}$$

$$\vec{T}_{model} = \frac{\vec{T}_{right} \cdot I_l}{I_l + I_r} + \frac{\vec{T}^r_{left} \cdot I_r}{I_l + I_r}, \quad if\ I_r, I_l > I_{min} \tag{3}$$

where I_l and I_r are the number of inliers from the left and right pose estimations, as determined with RANSAC. $\vec{R}_{model}$ and $\vec{T}_{model}$ are the resulting pose estimation. $\vec{R}_{right}$ and $\vec{T}_{right}$ are the pose estimation from the right image, and $\vec{R}^r_{left}$ and $\vec{T}^r_{left}$ are pose estimation from the left camera, translated to the right camera using the corresponding stereo equations and camera calibration parameters. In case the number of inliers of any of the cameras is less than the I_{min} threshold, set to half the total number of points, that estimation is discarded and the estimation of the other camera is used. If inliers for both images are below the threshold, the frame is rejected and the estimation from the previous frame is used. The final values of the pose are filtered with a Kalman Filter to smooth the response.

E. Feature point tracking failure detection and recovery

Points identified as outliers by the RANSAC algorithm are moved to a corrected position, so they can be tracked on the following frames. The new position of the points is calculated by re-projecting the 3D model on both camera planes with the final estimated pose, $\vec{R}_{model}$ and $\vec{T}_{model}$.

The SMAT model is also inspected when outliers are found, as it has been updated with all the image patches, regardless of their validity. Incorrect patches could contaminate the model and induce further tracking errors. Thus, patches that correspond to outlier points on the last frame are also considered outliers, and removed from the SMAT model.

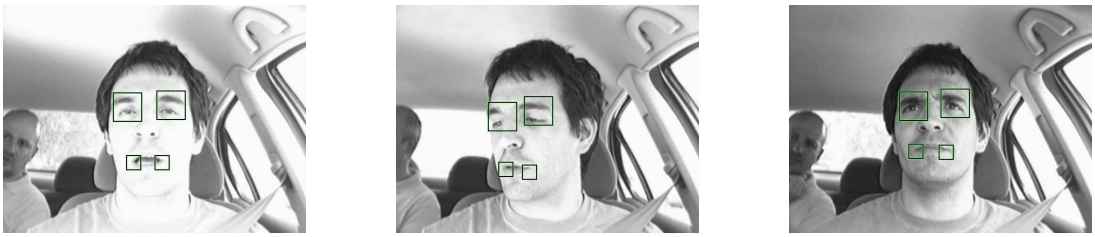


Fig. 5. SMAT based tracking

IV. Test setup and results

The algorithm has been tested with videos recorded in a moving car during daytime. In the videos, the driver faces front to the cameras on the first frames. The sequences recorded in the car show the subjects performing normal driving gestures and head movements, and they frequently talk. The videos were recorded using two synchronised FireWire cameras, at a framerate of 20 fps and a resolution of 960 x 480 pixels. Eight drivers participated in the recordings, of which 6 were male and 2 female. Four of them wore glasses. The lenght of each video sequence depends on the driver and traffic conditions, ranging from 2 minutes to 5 minutes. Several sequences were recorded for each driver. The total length of the sequences is over an hour. Most sequences were recorded in the streets of the University Campus at daytime, with the rest recorded in a urban environment. The weather conditions were mostly sunny, which made noticeable shadows appear on the half of the face further away from the window (see Figs. 6 and 8). Global illumination changes took place as the car moved, due to the presence of trees by the road. Local illumination changes affecting only part of the face occurred when the driver's head moved closer or further away from the window.

Assessing the performance of the approach requires obtaining a ground-truth value for the orientation of the head on each frame. Hand-marking several points over the face on every frame is a very time consuming procedure, and it is not error-free as there is always an error in the precision of the human operator and there may be deformations on the selected points too. We have tried to minimise the error on the ground-truth value by recording videos were the subjects wear a bicycle helmet where two printed chessboards have been attached, as can be seen in Fig. 6(a). This chessboards are a cut-out version of the ones commonly used for camera calibration, so the corner localisation techniques used in calibration could be used to determine it position and orientation. The chessboards were placed at the back of the helmet so they had a minimal impact on the subject movements and the helmet was still comfortable to wear, while being visible from at least one of the cameras. As with the driver face model, the position of the chessboards is determined on the first frame, and they are subsequently tracked for the whole duration of the recording. The corners on the chessboard are roughly tracked with the SMAT algorithm, and the fine position is obtained using the gradients of the squares' borders. The resulting videos were inspected for errors in the corner detection.

Subjects were also recorded driving without the helmet, and asked to drive the same streets. These videos do not have a ground truth value available, but demonstrate the performance of the approach when drivers are not wearing any additional equipment, as can be seen in Fig. 6(b). A few videos of the approach working on both types of set-ups can be found on our website[1].

The model construction step is performed on the first frame. The system chooses up to 30 characteristic tracking points to built the 3D model. Point positions are corrected, erroneous points are automatically removed, and the point occlusion pattern is created, based on a cylinder-like face.

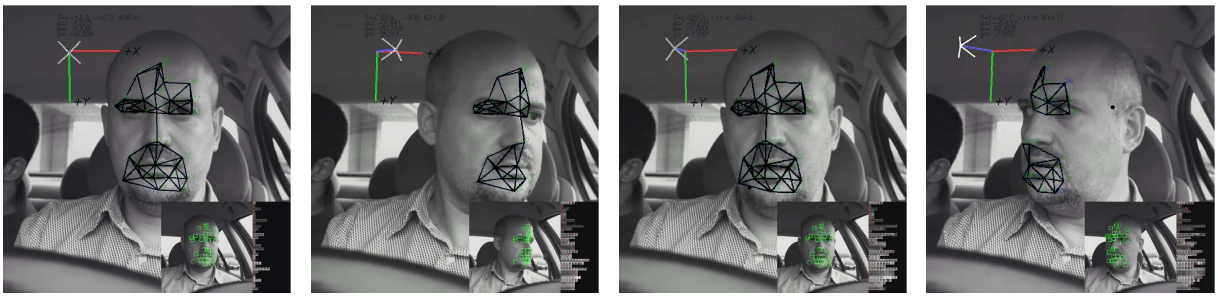
Fig. 7 shows the process of model creation, tracking and pose estimation. Pose is correctly estimated over face rotations. The more the face is rotated, the more points are hidden, and thus the accuracy of the pose estimation falls. This reduced accuracy appears for angles that result in more than 50% of the points being hidden (over ± 35 degrees). When approximately 75% of the model's points are hidden, the RANSAC algorithm does not have enough points to get the correct set of inliers and outliers, and the pose estimation fails to produce a value. The images cover different head rotations, and show the estimated pose vector. Fig. 8 shows frames from another test.

In most cases when rotations go over $\pm 50°$, track-

[1] http://robesafe.com/tecnologias/index_en.php

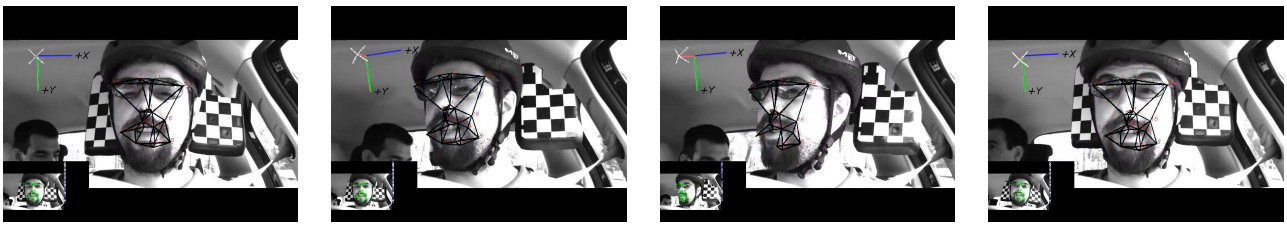
(a) Driver wearing a helmet with chessboards attached (b) Subject driving without helmet

Fig. 6. Shots from videos with and without helmet



(a) Frame 10 (b) Frame 80 (c) Frame 130 (d) Frame 225

Fig. 7. Model creation, tracking and pose estimation of the face of a driver



(a) Frame 75 (b) Frame 117 (c) Frame 271 (d) Frame 589

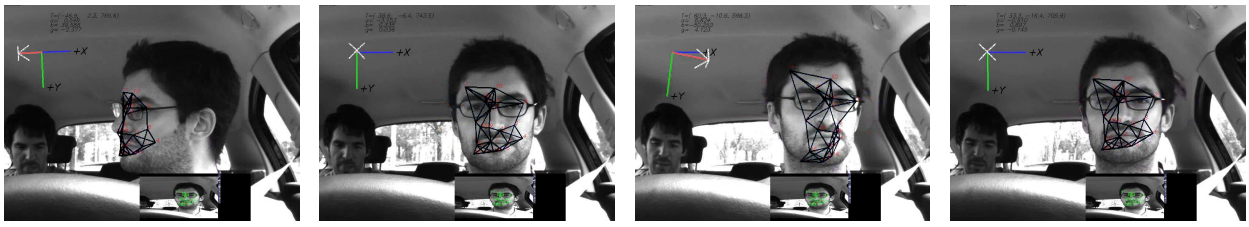
Fig. 8. Face tracking of a subject wearing the helmet

ing is lost. The system then searches the image for a frontal view of the face, first with the Viola&Jones algorithm, and then with the model patches in the area where the face has been found. Fig. 9 shows a few frames of a sequence where tracking is lost and the face is found again. Tracking recovery is very fast, and pose parameters are correctly estimated again in less than 0.5 seconds (10 frames).

Figs. 10 and 11 show the estimated and ground-truth values for two videos of different drivers. As can be seen, both values are very close and frequently overlap. Estimation error stays in acceptable values in presence of severe rotations. Combined rotations are correctly handled, as happens around frame 500 in Fig. 10, where a remarkable turn occurs for the pitch and yaw angles. Illumination changes were also correctly handled.

The driver in the video corresponding to Figs. 8 and 11 talks frequently and rises his eyebrows in different moments. It can be seen that the error values remain very similar throughout the video, with no significant differences due to the presence of facial expressions. As mentioned above, when the magnitude of the expressions is small the fixed-body assumption holds for most of the points and the introduced errors can be corrected with RANSAC+POSIT.

Table I shows the Mean Absolute estimation error of the proposed system along with three other methods referenced above [23], [6], [26] that are representative approaches to the problem of head pose estimation. The error values for our system are for frames where face tracking is not lost, as in the event of a loss an estimation can not be obtained. The comparison should be taken with caution, because



(a) Frame 584 (b) Frame 600 (c) Frame 640 (d) Frame 660

Fig. 9. Tracking loss and recuperation.

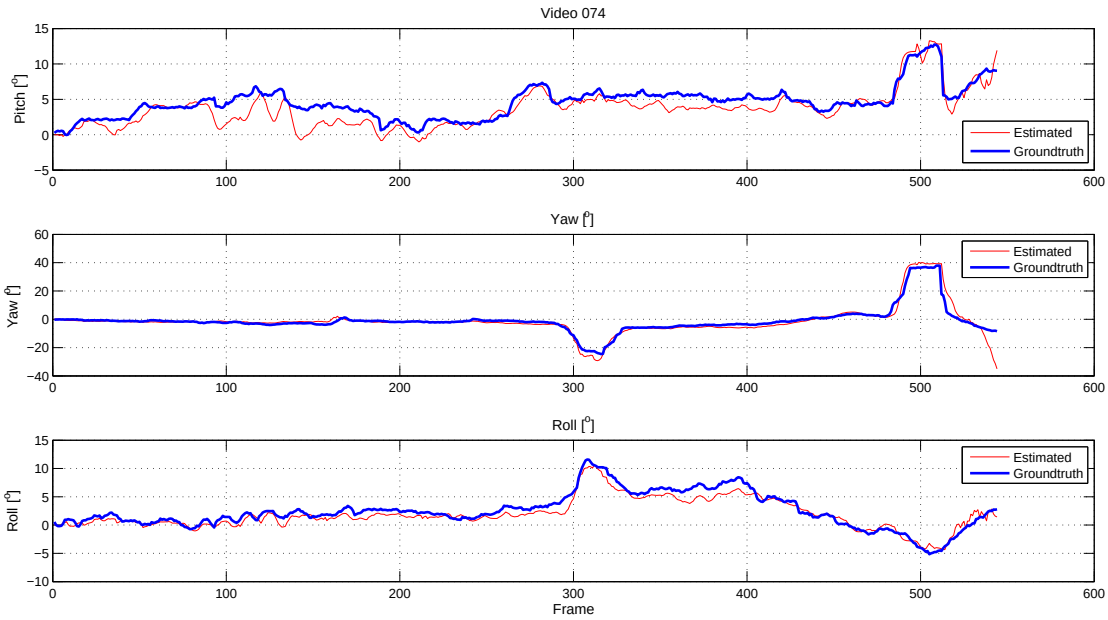


Fig. 10. Estimated and ground-truth pose for video 074

the datasets used to evaluate the methods and the systems used to obtain the ground-truth values are all different. Some methods use monocular systems and other use stereo camera setups. In terms of the video datasets, Murphy-Chutorian and Trivedi's and the one used in this work are the biggest, and the only ones recorded in a moving vehicle. Both contain sequences recorded at daytime, while the former also includes sequences recorded at night. The error values of the proposed system are in line with the best methods presented to date, with the added benefit that it does not require an off-line training step.

For the proposed system, the highest error values are those for the yaw angle. This is expected, as the rotations of the subjects' head along this angle are more pronounced than along the other angles, and the accuracy of the estimation is lower for angles over $\pm 35°$. Tracking losses also increased with the magnitude of the rotation. Table II shows the percentage of head turns in the yaw angle that led to tracking losses.. The error figures are consistent over different videos and users.

TABLE II
PERCENTAGE OF HEAD TURNS THAT LED TO TRACKING LOSSES

Yaw angle range	$\lvert\beta\rvert \leq 35°$	$35° < \lvert\beta\rvert \leq 45°$
Tracking losses	4%	32%

The algorithm has been coded in C/C++ using the OpenCV Library, and it is able to run in real-time in a 2.4GHz Core2 Duo processor. Table III shows the mean and maximum processing times for tracking and pose estimation, for the given system with 30 points. Tracking with SMAT is the most time consuming process. Processing times vary slightly depending on the number of iterations required, but they are below the real-time threshold in all our tests.

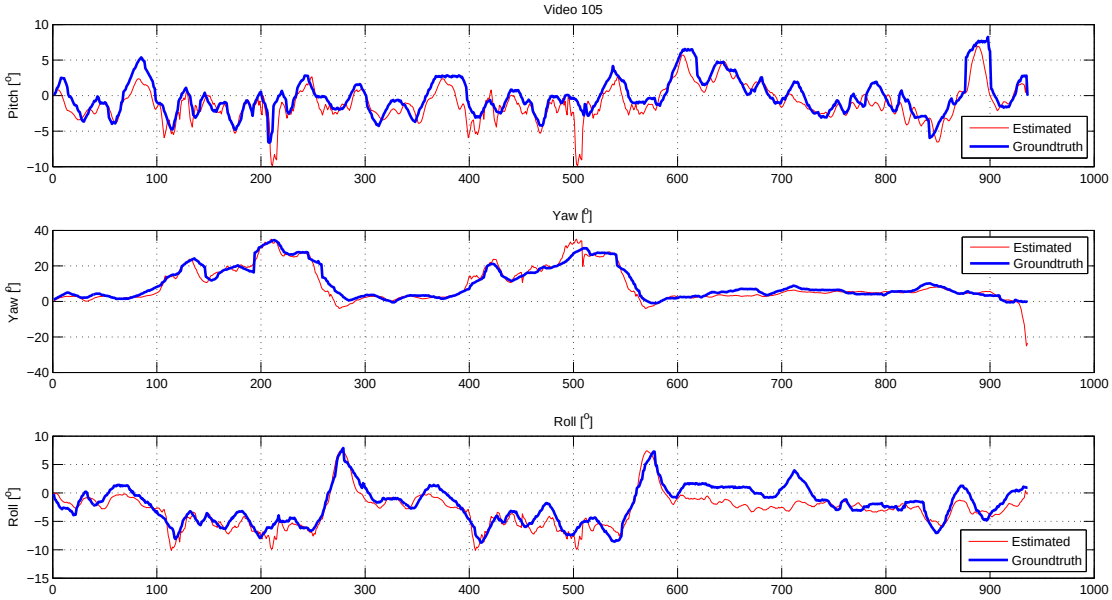


Fig. 11. Estimated and ground-truth pose for video 105

TABLE III
PROCESSING TIMES

Task	Mean time	Max time
SMAT	18 ms	21 ms
RANSAC+POSIT	13 ms	15 ms

V. CONCLUSIONS

This paper presents a robust face tracking and pose estimation method using stereo vision that runs in real-time. Our approach automatically constructs a 3D model with feature points located on the face, just requiring the user to look straight ahead for a few frames. Tracking of feature points is carried out separately on left and right images using SMAT, and incorrectly tracked points are rejected using RANSAC. 3D pose is recovered from the 2D point set using POSIT.

The algorithm has been tested in video sequences recorded in a moving vehicle, and works reliably for face rotations under $\pm 45°$, with mean absolute estimation error below $2°$. Rotations greater than this value result in a great number of points being occluded, and the pose can not be estimated. We are working towards on-line model extension to solve this problem. This would augment the model when the face pose is extreme, and the algorithm accuracy drops below a threshold due point occlusion. A modification of the technique used to create the initial frontal model will be used to extend it. We will also study different statistics on the tracking results for each 3D point, so points that can not be consistently tracked are removed from the model. Finally, we plan to analyse the system performance with lower resolution images. With these additions, this algorithm is to be used as the base of a inattention monitoring system for drivers.

ACKNOWLEDGEMENT

This work has been funded by grants `TRA2005-08529-C02-01` (MOVICOM Project) and `PSE-370100-2007-2` (CABINTEC project) from the Spanish Ministry of Science and Technology (MCyT), as well as S-0505/DPI/000176 (Robocity2030 Project) from the Science Department of the Comunidad de Madrid. J. Nuevo is also working under a researcher training scholarship from the Education Department of the Comunidad de Madrid and the European Social Fund. The authors would like to thank the drivers that collaborated in the recording of the test sequences.

REFERENCES

[1] M. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, 2002.

TABLE I
Pose Estimation Mean Absolute Error

Method	Configuration	Mean Absolute Error Yaw	Pitch	Roll	Dataset	Working angle range
La Cascia[23]	Monocular	3.0°	6.1°	9.8°	A	Up to ±60°
Murphy-Chutorian[6]	Monocular	3.39°	4.67°	2.38°	B	Full
Morency[26]	Stereo	3.5°	2.4°	2.6°	C	Full
Proposed system	Stereo	1.85°	1.61°	1.2°	D	Up to ±45°

Datasets:

A) 72 video sequences recorded in a laboratory with 9 subjects, with uniform and time varying illumination. All sequences are 200 frames long. Rotation angles up to 60°. Ground-truth was recorded with a magnetic sensor. Available online at http://www.cs.bu.edu/groups/ivc/HeadTracking.

B) 14 video sequences of a subject while driving in daytime and nighttime (with near-IR illumination). Each sequence is approximately 8-minutes. Head motion range is fully covered. Ground-truth recorded with an optical motion capture system. The dataset is not available.

C) One 800-frame video sequence of a user at 7 Hz. Head motion range is fully covered. Ground-truth was collected with an inertial sensor. Dataset is not available.

D) 20 sequences of eight subjects driving a vehicle, of 2 to 5 minutes in length. Head motion range is fully covered. Ground-truth recorded by tracking patterns attached to a helmet. Dataset is not available.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, July 1997.

[3] J. Buenaposada, E. Muñoz, and L. Baumela, "Recognising facial expressions in video sequences," *Pattern Analysis & Applications*, vol. 11, no. 1, pp. 101–116, 2008.

[4] F. Dornaika and F. Davoine, "Head and Facial Animation Tracking using Appearance-Adaptive Models and Particle Filters," in *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, 2004, pp. 153–153.

[5] R. Stiefelhagen, J. Yang, and Waibel, "Tracking eyes and monitoring eye gaze," in *Proceedings of PUI'97*, 1997.

[6] E. Murphy-Chutorian, A. Doshi, and M. Trivedi, "Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation," in *Proc. of Intell. Transport. Syst. Conf. 2007*, 2007, pp. 709–714.

[7] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and E. López, "Real-time system for monitoring driver vigilance," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 1524–1538, Mar. 2006.

[8] R. Senaratne, D. Hardy, B. Vanderaa, and S. Halgamuge, "Driver Fatigue Detection by Fusing Multiple Cues," *Lecture Notes In Computer Science*, vol. 4492, p. 801, 2007.

[9] A. Gee and R. Cipolla, "Determining the gaze of faces in images," *Image and Vision Computing*, vol. 12, no. 10, pp. 639–647, 1994.

[10] W. Liao and G. Medioni, "3D face tracking and expression inference from a 2D sequence using manifold learning," *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, 2008.

[11] M. Breitenstein, D. Kuettel, T. Weise, L. van Gool, and H. Pfister, "Real-time face pose estimation from single range images," *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conf. on*, pp. 1–8, June 2008.

[12] A. Kapoor and R. Picard, "Real-time, fully automatic upper facial feature tracking," *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pp. 8–13, 2002.

[13] Z. Zhu and Q. Ji, "Robust real-time face pose and facial expression recovery," *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 681–688, 2006.

[14] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-time combined 2D+ 3D active appearance models," in *IEEE Conference on Computer Vision and Pattern Recognition 2004*, vol. 2, 2004, pp. 535–542.

[15] J. Yao and W. Cham, "Efficient Model-Based Linear Head Motion Recovery from Movies," *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, vol. 2, 2004.

[16] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, pp. 681–685, Jan. 2001.

[17] L. Gu and T. Kanade, "3D Alignment of Face in a Single Image," *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1*, pp. 1305–1312, 2006.

[18] 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.

[19] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vision*, vol. 15, no. 1-2, pp. 123–141, 1995.

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] P. Jimenez, J. Nuevo, and L. Bergasa, "Face pose estimation and tracking using automatic 3d model construction," *CVPR Workshops, 2008. IEEE Computer Society Conference on*, pp. 1–7, June 2008.

[22] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, April 2009.

[23] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models,"

IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 4, pp. 322–336, 2000.

[24] T. Cootes, G. Wheeler, K. Walker, and C. Taylor, "View-based active appearance models," *Image and Vision Computing*, vol. 20, no. 9-10, pp. 657–664, 2002.

[25] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1063–1074, 2003.

[26] L. Morency, P. Sundberg, and T. Darrell, "Pose estimation using 3D view-based eigenspaces," in *Analysis and Modeling of Faces and Gestures, AMFG 2003. IEEE International Workshop on*, 2003, pp. 45–52.

[27] J. Pilet, V. Lepetit, and P. Fua, "Real-time non-rigid surface detection," in *IEEE Conference on Computer Vision and Pattern Recognition 2005*, San Diego, CA, June 2005.

[28] M. Ozuysal, P. Fua, and V. Lepetit, "Fast keypoint recognition in ten lines of code," in *IEEE Conference on Computer Vision and Pattern Recognition 2007*, Minneapolis, MI, June 2007.

[29] Z. Zhu and Q. Ji, "Real Time 3D Face Pose Tracking From an Uncalibrated Camera," *CVPR Workshop 2004, Conference on*, pp. 73–80, 2004.

[30] C. Harris and M. Stephens, "A combined corner and edge detector," *Alvey Vision Conference*, vol. 15, p. 50, 1988.

[31] P. Viola and M. Jones, "Robust real-time object detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2002.

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

[33] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.