UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Electrónica

Face Pose Estimation with Automatic 3D Model Creation for a Driver Inattention Monitoring Application

A Thesis submitted for the degree of Doctor of Philosophy

Author

Pedro Jiménez Molina

Supervisor

Dr. D. Luis Miguel Bergasa Pascual

2011

Agradecimientos

Si bien un modelo incremental e iterativo puede ser bueno para muchas aplicaciones y tal vez para el desarrollo de una tesis doctoral, desde luego no lo es tanto para los que tienen la tarea de corregirla, de forma igualmente incremental e iterativa. Es por ello que empiezo agradeciendo la encomiable labor de corrección que han realizado el Dr. Luis Miguel Bergasa y el Dr. Jesús Nuevo.

A Luis Miguel por su constante ayuda y apoyo. Sin él, esta tesis corría el riesgo de no haber terminado, y tampoco hubiera evolucionado como lo ha hecho. Le tengo que agradecer enormemente su dedicación e infinita paciencia ya desde los comienzos del doctorado, hace ya más de cuatro años.

A Jesús le agradezco sus continuos mensajes de ánimo. Su trabajo sirvió de base para mis investigaciones, y su ayuda con la lengua de Shakespeare ha probado ser inestimable. En definitiva, Jesús ha sido imprescindible para escribir una tesis fuerte y sana. Desde el año 2000 con nuestro primer robot, ha estado ahí compartiendo proyectos y trabajando de más por mi culpa.

Al resto de miembros de RobeSafe, que hacen las horas de trabajo y desarrollo mucho más llevaderas, también agradecerles su amistad, simpatía y apoyo constante. A Pablo, Llorca, Sotelo, Nacho, Revenga por su interminable ayuda teórica y técnica. A Iván, los alcarreños Edu y Raul, y el último en llegar, pero no menos importante, Almazán, por ser tan buenos compañeros, y poner ese grado de humanidad y ¿cordura? necesario en todo laboratorio. A Javi por contribuir con la causa y venir a currar en bici. A Sergio por organizar los partidos de fútbol. Y también al resto de compañeros de laboratorio, Álvaro, Ángel, Carlos, David, Fer, Miguel, Noe y Óscar por su buen humor.

En el desarrollo técnico y testeo también ha sido imprescindible la ayuda de los miembros del Consorcio CABINTEC, Beatriz Delgado y Matías Sevillano de ESM, y el resto de miembros del CEIT e instituciones implicadas: Universidad Rey Juan Carlos y Universidad de Valencia.

Quisiera agradecérselo de todo corazón también a mi familia más cercana: mis padres, mi hermano Santi, mis abuelos que no me ven el pelo, y por supuesto a Loli, que ha dejado de hacer tantas cosas por culpa de la que es nuestra primera tesis (chispas), y que ha acabado por aprendérsela de tanto releerla. Gracias a ella los años de trabajo han pasado más rápido y felices. A mi hermano Santi por sacarme de paseo a hacer un poco de ejercicio, ya que la tesis ayuda a la mente (o no), pero oxida las articulaciones. Gracias a mi madre que tantas veces me ha esperado para comer y yo mientras escribía la tesis olvidé la cita. A mi padre, al cual no he podido visitar tan a menudo como me hubiera gustado, y que ha sabido entenderlo. Y aunque ella todavía no sea consciente, agradecer también a mi hermana Paula, por hacerme recordar lo feliz que yo era de pequeño, y por ayudarme a ver lo feliz que ahora soy también. Paula, como el resto, me ve con poca frecuencia, pero ella es la única que aún no comprende porqué. Y gracias también a Loli y Diego, pues porque se lo merecen. Me temo que, al igual que mi madre, todavía no

tienen muy claro de qué va la tesis (y así seguirá siendo, por culpa del inglés) o para qué sirve, pero eso no ha cambiado su postura de apoyarme y animarme incondicionalmente desde el principio.

También quisiera mandar un profundo agradecimiento a todos mis amigos, a los de Alcalá y a los de Guada, que estaban ahí antes de la tesis, y siguen estando después, a pesar de las pocas visitas intermedias.

Por otro lado, la tesis también a aportado viajes inolvidables como las estancias rodeado de Leprechauns y en Down Under. Algunos de los mejores momentos durante estos más de cuatro años tuvieron lugar en la tierra de Oz, como el atravesar zonas boscosas cada día mientras iba a trabajar en bicicleta. Todo ello compensó el esfuerzo de las reuniones a través de Skype.

En resumen, son muchas las personas a las que pertenece una pequeña o gran parte de este trabajo, pero como aún son más las ganas de acabar, los agradecimientos finalizan aquí.

Gracias a todos.

"Los hemos visto. Hemos encontrado a los hombres." (...) *"Caminan sobre cuatro extremidades en forma de ruedas que giran vertiginosamente."*

De la Tierra a Halley, Lucía Baquedano, 1988.

"We have seen them. We have found men." (...) *"They walk on four extremities in the shape of wheels that run in a giddy way."*

From the Earth to Halley, Lucía Baquedano, 1988.

Resumen

Recientes estudios han identificado la inatención (incluyendo distracción y somnolencia) como la mayor causa de accidentes, siendo responsable de al menos un 25% de ellos. La distracción en conductores se ha estudiado menos, ya que depende de muchos factores, aunque representa un mayor riesgo que la fatiga. Además, la distracción está presente en más de la mitad de los accidentes causados por algún tipo de inatención. Cada día existen más sistemas de información embarcados en los vehículos (*In Vehicle Information Systems*, IVIS), lo que incrementa el riesgo de provocar distracciones y modifica el comportamiento de los conductores. Esto hace que las investigaciones en este ámbito sean de vital importancia.

Para abordar el análisis de las distracciones durante la conducción, distintos grupos de investigadores han trabajo en diversas técnicas, entre las que destaca la Visión por Computador dado que permite, mediante el uso de tecnología relativamente barata, la monitorización del conductor de forma no intrusiva. Mediante técnicas de visión como el seguimiento facial se puede evaluar su movimiento con objeto de caracterizar el estado de atención del conductor.

En esta tesis se presentan varias técnicas de visión 3D usando una cámara estéreo para obtener en tiempo real y de forma completamente automática la dirección de la cara y de la mirada de una persona. A partir de esta información se infieren las distracciones en el conductor. Los métodos aquí mostrados funcionan de forma completamente automática e independiente del usuario.

Para detectar la dirección de la cara del conductor, primero se crea un modelo 3D no denso usando las coordenadas de puntos característicos de la misma, obtenidos gracias al par de cámaras estéreo. Durante la ejecución del algoritmo, se hace un seguimiento de los puntos característicos, mientras el modelo se va ampliando y corrigiendo automáticamente cuando nuevas partes de la cara, previamente ocultas, se hacen visibles a las cámaras.

Se evalúan varias técnicas para la determinación y seguimiento de los puntos del modelo. Primeramente se estudia el comportamiento de un seguidor basado en descriptores SURF, por ser una de las técnicas más ampliamente usadas en visión. Sin embargo, debido a las condiciones de baja iluminación y lo suaves que son los contornos de una cara, esta técnica no produce buenos resultados. Este hecho, unido al elevado coste computacional de la misma, hacen que dicha técnica sea descartada. Por ello, se diseña una técnica de seguimiento mediante correlación *multisize* (multitamaño), basada en el uso de parches de distintos tamaños a una misma escala. Esta técnica ofrece una leve mejora en el posicionamiento y tiempos de ejecución con respecto al uso de parches multiescala. Esta técnica es robusta gracias a la aportación de los parches más grandes, y es de más precisión gracias a los parches más pequeños.

La cara puede rotar en un rango horizontal de $\pm 90°$, lo que hace que la apariencia de los puntos característicos cambie notablemente. Para abordar este problema, se introduce una técnica novedosa de *re-registering* para robustecer el seguimiento de las caracterís-

ticas que forman el modelo, aprovechando las vistas que se tienen de la cara desde las distintas cámaras. La muestra de cada característica que se tiene almacenada y se usa para la localización del punto 2D sobre la cara se va actualizando conforme la cara rota, aprovechando los puntos de mínimo error en la estimación de la pose. De este modo, cada muestra solo se usa en el *tracking* en un rango de $\pm 7,5°$.

Puesto que el modelo se crea inicialmente con una vista frontal de la cara, solo se pueden capturar puntos característicos de la parte frontal. Cuando se producen rotaciones, algunos de esos puntos se ocultan, por lo que se hace necesario añadir nuevos puntos al modelo para evitar que el número de puntos visibles disminuya. Tras añadir puntos de partes de la cara previamente ocultas, se ejecuta un *Bundle Adjustment* para reducir el error acumulativo que se puede producir al añadir puntos.

El modelo 3D de la cara sirve de apoyo para reconstruir la posición 3D de la misma usando uno de los dos algoritmos evaluados, bien sea POSIT o Levenberg-Marquardt, siendo el primero más rápido, y LM más preciso. Además, un proceso RANSAC permite detectar puntos incorrectos o *outliers*, y descartarlos para la estimación de la pose. Gracias a la unión de todos los métodos mencionados, se consigue un sistema de seguimiento que funciona en el rango completo de rotación de la cara, y que mejora los resultados del estado del arte.

A la estimación de la pose de la cara se añade una estimación de la dirección de la mirada y del punto de focalización de la misma. Estos datos aportan gran información sobre el comportamiento del conductor y su grado de distracción.

En el desarrollo de la tesis se evalúan y comparan las distintas técnicas mencionadas, usando para ello una extensa colección de vídeos. El algoritmo de estimación de la mirada propuesto en esta tesis se valida mediante un conjunto de experimentos de conducción en un simulador realista, definidos por un equipo de psicólogos. Se han simulado cambios climatológicos, maniobras y distracciones debidas a IVIS. Las pruebas han sido realizadas por conductores profesionales.

Los resultados estadísticos obtenidos sobre la fijación de la mirada muestran cómo la utilización de IVIS influye en el comportamiento de los conductores, incrementando sus tiempos de reacción y afectando a la fijación de su mirada sobre la carretera y sus alrededores.

Abstract

Recent studies have identified inattention (including distraction and drowsiness) as the main cause of accidents, being responsible of at least 25% of them. Driving distraction has been less studied, since it is more diverse and exhibits a higher risk factor than fatigue. In addition, it is present over half of the inattention involved crashes. The increased presence of *In Vehicle Information Systems* (IVIS) adds to the potential distraction risk and modifies driving behaviour, and thus research on this issue is of vital importance.

Many researchers have been working on different approaches to deal with distraction during driving. Among them, Computer Vision is one of the most common, because it allows for a cost-effective and non-invasive driver monitoring and sensing. Using Computer Vision techniques it is possible to evaluate some facial movements that characterise the state of attention of a driver.

This thesis presents methods to estimate the face pose and gaze direction of a person in real-time, using a stereo camera as a basic for assessing driver distractions. The methods are completely automatic and user-independent. A set of features in the face are identified at initialisation, and used to create a sparse 3D model of the face. These features are tracked from frame to frame, and the model is augmented to cover parts of the face that may have been occluded before. The algorithm is designed to work in a naturalistic driving simulator, which presents challenging low light conditions.

We evaluate several techniques to detect features on the face that can be matched between cameras and tracked with success. Well-known methods such as SURF do not return good results, due to the lack of salient points in the face, as well as the low illumination of the images. We introduce a novel *multisize* technique, based on Harris corner detector and patch correlation. This technique benefits from the better performance of small patches under rotations and illumination changes, and the more robust correlation of the bigger patches under motion blur.

The head rotates in a range of $\pm 90^\circ$ in the yaw angle, and the appearance of the features change noticeably. To deal with these changes, we implement a new re-registering technique that captures new textures of the features as the face rotates. These new textures are incorporated to the model, which mixes the views of both cameras. The captures are taken at regular angle intervals for rotations in yaw, so that each texture is only used in a range of $\pm 7.5^\circ$ around the capture angle. Rotations in pitch and roll are handled using affine patch warping.

The 3D model created at initialisation can only take features in the frontal part of the face, and some of these may occlude during rotations. The accuracy and robustness of the face tracking depends on the number of visible points, so new points are added to the 3D model when new parts of the face are visible from both cameras. Bundle adjustment is used to reduce the accumulated drift of the 3D reconstruction.

We estimate the pose from the position of the features in the images and the 3D model using POSIT or Levenberg-Marquardt. A RANSAC process detects incorrectly tracked

points, which are not considered for pose estimation. POSIT is faster, while LM obtains more accurate results. Using the model extension and the re-registering technique, we can accurately estimate the pose in the full head rotation range, with error levels that improve the state of the art.

A coarse eye direction is composed with the face pose estimation to obtain the gaze and driver's fixation area, parameter which gives much information about the distraction pattern of the driver. The resulting gaze estimation algorithm proposed in this thesis has been tested on a set of driving experiments directed by a team of psychologists in a naturalistic driving simulator. This simulator mimics conditions present in real driving, including weather changes, manoeuvring and distractions due to IVIS. Professional drivers participated in the tests.

The driver's fixation statistics obtained with the proposed system show how the utilisation of IVIS influences the distraction pattern of the drivers, increasing reaction times and affecting the fixation of attention on the road and the surroundings.

Contents

List of Figures

List of Tables

Notation

Face pose

$\mathbf{X}_i = (x_i, y_i, z_i)$	: Coordinates in $\mathbb{R}^3$ of the 3D feature point i, in word coordinates
N_0	: Initial number of points 3D points in the face model
N	: Number of points 3D points in the face model after any extension
$\mathbf{X}_i^{(\mathcal{M})} = (x_i, y_i, z_i)$	: Coordinates in $\mathbb{R}^3$ of the 3D face model feature point i in object coordinates
$\mathcal{M} = \{\mathbf{X}_i^{(\mathcal{M})}\}_{i=1...N}$	: 3D face model and set of 3D points which from the model, in object coordinates
$\mathbf{P}$	: $(V_x, V_y, V_z, T_x, T_y, T_z)$ 6 DoF parametric pose vector of the face, composed of the rotation vector $\mathbf{V}$ and translation $\mathbf{T}$
$\{R, T\}$	: Rotation and translation matrices, representing the 3D face model pose
R'	: 3×3 corrected Rotation matrix to diminish the translation effect
$I_t^{\{r,l\}}$	: Right/Left camera images, at frame t
$I_0^{\{r,l\}}$	: Right/left camera images at model creation, frame $t = 0$
$\mathbf{x}_{i,t}^r = (u_i^r, v_i^r)$	: ($\mathbb{R}^2$) Projection on I^r of the 3D face model point $\mathbf{X}_i$
$\mathbf{x}_{i,t}^l = (u_i^l, v_i^l)$	: ($\mathbb{R}^2$) Projection on I^l of the 3D face model point $\mathbf{X}_i$
$\mathbf{T}^{(r)}$	: Template descriptor of feature i on image I^r, or texture for patch correlation
$\mathbf{T}_i$	: Model stored texture of feature i
$\mathbf{C}_i = \{\mathbf{T}_i^{(j)}\}_{j=0,1,...}$	: Cluster of model stored textures for feature i
P_i'	: Warped patch $P_i \rightarrow P_i^1$
$P_{i,t}^r$	: Patch in the right camera image I_t^r
$P_{i,t}^l$	: Patch in the right camera image I_t^l
$h_i^r = (u_i^r, v_i^r)$	: Coordinates in $\mathbb{R}^2$ of a feature candidate i on I_0^r
$h_i^l = (u_i^l, v_i^l)$	: Coordinates in $\mathbb{R}^2$ of a feature candidate i on I_0^l
n'	: Number of feature candidates in the face model
$P_{ik}^r = P(h_i^r, \mathbf{s}_k)$	: Patch template around projection point h_i^r on $I^r$and size $\mathbf{s}_k = s_{kx} \times s_{ky}$
$r_{i,k}^l(u, v)$	: Correlation results on I^lafter correlating patches of size $\mathbf{s}_k = s_{kx} \times s_{ky}$
$r_i^l(u, v)$	: Matching result on I^l, by composing all the $r_{i,k}^l(u, v)$
$\mathbf{s}_{search}$	: $(x_{search} \times y_{search}) \in \mathbb{R}^2$. Search area for a feature within the image

Gaze

$\vec{g}$	: Unitary gaze vector
T_g	: Point of origin of the gaze vector
$\mathbf{G} = \{T_g, \vec{g}\}$	: Gaze estimation (Origin and direction)
$\vec{e}$	: Eyes direction in model coordinate frame system
$\vec{e}_{off}$	: Offset to eyes' centre with respect to the model's centre m_o, in model coordinate frame

Chapter 1

Introduction

1.1 Motivation

Since Computer Vision beginnings, one of first challenges that researches have been trying to solve is the Interactive human-machine interfacing. Face pose and the focus of attention include a lot of information on human non-verbal language. People have the innate ability to detect the orientation of a human head, and easily capture the significant and non-verbal communication contained in these movements. Providing a robot with such ability has been and still is an intensive field of study. The face pose gives a lot of information in a communication process, on one hand in the intended movements, such as focusing attention on a area or looking at someone, and in the other hand in the unintended movement, such as face down while talking to other person. For a robot to be able to naturally interact to a person, it first needs to be able to estimate her face pose and her gaze. Gaze estimation can provide a user-machine interface with higher and better user experience, since communication will no be limited to manual interaction.

Another important field for the face pose estimation systems is the automotive industry. Since its early start, manufacturers have always dedicated plenty of resources to innovation, leading to many advances in the automotive technology. Better and more efficient engines, reduced production cost and more comfort lead to an important increase in the number of vehicles circulating. This situation has increased demands for safety and manufacturers are now focusing more and more on vehicle safety.

Driving inattention is a major factor to traffic crashes. In the EU-27, 42,854 people died in 2007 in traffic accidents [UN-ECE 07], and 44,400 people lost their lives in 2006 [Mahieu 09]. That year, over 1.25 million accidents took place and more than 1.5 million people were injured [SafetyNet 08]. Inattention has been found to be involved in some form in 80% of crashes and 65% of the near crashes within 3 seconds of the event [Dingus 06]. In an effort to reduce these figures, the European Commission set up in 2003 the European Road Safety Action Programme (2003-2010) [EC 03], which aims to halve the number of victims in road accidents by 2010. On the other hand, the *National Highway Traffic Safety Administration* (NHTSA) estimates that approximately 25% of police-reported crashes involve some form of driving inattention [Ranney 01]. The study of *American Automobile Association Foundation for Traffic Safety* (AAA FTS) showed the driving attention status has five categories: attentive, distraction, cognitive distraction (looking without seeing), fatigue and unknown [Ranney 01]. In this thesis, we will focus on the distraction category.

Driving distraction is defined by the AAA FTS as occurring "when a driver is delayed in the recognition of information needed to safely accomplish the driving task because

some event, activity, object or person within or outside the vehicle compelled or tended to induce the driver's shifting attention away from the driving task" [Young 07]. Thirteen types of potentially distracting activities are listed in [Stutts 01]: eating or drinking, outside person, object or event, talking or listening on cellular phone, dialling cellular phone, using in-vehicle-technologies, etc. Since the distracting activities take many forms, NHTSA classifies distraction into 4 categories from the view of the driver's functionality: visual distraction, cognitive distraction, auditory distraction (e.g., responding to a ringing cell phone), and biomechanical distraction (e.g., manually adjusting the radio volume) [Ranney 01]. Many distracting activities can involve more than one of these components (e.g., talking to a phone while driving creates a biomechanical, auditory and cognitive distraction). Driving distraction is more diverse and implies a more risky factor that fatigue and it is present in over half of inattention involved crashes, resulting in as many as 5000 fatalities and $40 billion in damages each year [Stutts 01]. Increasing use of *in-vehicle information systems* (IVIS) such as cell phones, GPS navigation systems, DVDs and satellite radios and other on-board devices has exacerbated the problem by introducing additional sources of distraction [Ranney 08]. Enabling drivers to benefit from IVIS without diminishing safety is an important challenge.

The purpose of a *Driver Inattention Monitoring Application* (DIMA) is to monitor the attention status of the driver. If driver inattention is detected, different countermeasures should be taken to maintain driving safety, depending on the types and levels of inattention. DIMA has been an active research field for decades. A large amount of scientific work has been done in this field, and various methods have been proposed. Some auto companies have already installed some simple function driver fatigue monitoring systems in their high-end vehicles. Yet, there is still a great need to develop a more reliable and fully functional DIMA, using cost efficient methods for a real driving context. It is believed that the development of signal processing and computer vision techniques will attract more attention to the study of this field in the coming years. With the intention of benefiting those interested in this field, this thesis gives a comprehensive review of the state of the knowledge on driver distraction. It thus provides a clear view of the previous achievements and the issues that still need to be considered.

1.2 Distraction effects on driving behavioural performance

Performing a cognitively demanding task while driving would influence the driver's visual and physiological behaviour and the driving performance.

1.2.1 Driver's visual behaviour

With an increase in the cognitive demand, many drivers change their inspection patterns on the forward view. [Angell 06] indicated that the eye-glance pattern could be used to discriminate driving while performing a secondary task from driving alone, and could be used to discriminate high- from low-workload secondary tasks. More facts associated with cognitive distraction driving can be found in [Harbluk 07] - [Rantanen 99]: drivers narrowed their inspection of the outward view and spent more time looking directly ahead. They reduced their inspection of the instruments and mirrors, and reduced their glances at traffic signals and the area around an intersection. [Rantanen 99] found that the visual field shrank by 7.8% during a moderate-workload counting task and by 13.6% during a cognitively demanding counting task. Drivers had fewer saccades per unit time, which was consistent with a reduction in glance frequency and less exploration of the driving

environment, and in some cases drivers shed these tasks completely and did not inspect these areas at all [Harbluk 02]. [Hayhoe 04] showed links between eye movement (fixation, saccade, and smooth pursuit), cognitive workload, and distraction. Fixations occur when an observer's eyes are nearly stationary. Saccades are very fast movements that occur when visual attention shifts from one location to another. Smooth pursuits occur when an observer tracks a moving object such as a passing vehicle. Saccade distance decreases as task complexity increases, which indicates that saccades may be a valuable index of mental workload [Greef 09]. In contrast, the amount of head movement increased when cognitive loads were imposed. It is believed to be a compensatory action by which a driver attempts to obtain a wider field of view [Miyaji 09]. Miyaji proposed that the standard deviations of eye movement and head movement could be suitable for detecting the states of cognitive distraction in subjects. Both cognitive and visual distractions caused gaze concentration and slow saccades when drivers looked at the roadway, and cognitive distraction increased blink frequency [Liang 10]. Liang and Lee found in their work that visual distraction resulted in frequent, long off-road glances. A report from the Safety Vehicle Using Adaptive Interface Technology (SAVEIT) program showed that eyes-off-road glance duration, head-off-road glance time, and Standard Deviation of Lane Position (SDLP) are good measures of visual distraction [Zhang 08].

1.2.2 Driver's physiological responses

When cognitive loads (conversation or arithmetic) were imposed on subjects, pupil dilation occurred by the acceleration of the sympathetic nerve [Miyaji 09]. The average heart rate also increased by approximately 8 beats per minute. However, the average value of the heart rate (RRI) decreased under the same situation. [Itoh 09] pointed out that performing a cognitively distracting secondary task (e.g., talking, thinking about something, etc.) during driving caused a decrease in the driver's temperature at the tip of the nose, and this effect was reproducible. It was reported in [Wesley 10] that a considerable and consistent skin temperature increase in the supraorbital region could be observed during cognitive and visual distractions. [Berka 07] found that the electroencephalography (EEG) signal also contained information about the task engagement level and mental workload.

1.2.3 Driving performance

Significant changes were observed in a driver's vehicle control as a consequence of performing the additional cognitive tasks while driving. [Ranney 08] found that distraction may be associated with lapses in vehicle control, resulting in unintended speed changes or allowing the vehicle to drift outside the lane boundaries. [Zhou 08] found the influences on the lane changing behaviour when a secondary task was being performed, which included a reduction in the frequency of the checking behaviour (check a side mirror or speedometer), a delay in the checking behaviour, and a longer time for the checking behaviour. [Carsten 05] found that the effects of cognitive distraction on driving performance differed considerably from those of visual distraction. Visual distraction affects a driver's steering ability and lateral vehicle control, while cognitive distraction affects longitudinal vehicle control, particularly car-following. [Liang 10] also found that cognitive distraction made steering less smooth, but improved lane maintenance. According to them, steering neglect and overcompensation are associated with visual distraction, while under-compensation is associated with cognitive distraction. Overall, visual distraction interferes with driving

performance more than cognitive distraction. An apparently anomalous finding is that when secondary task cognitive demands increased, a driver's lateral control ability was found to improve [Carsten 05]. [Harbluk 07, Harbluk 02] related an increased incidence of hard braking associated with cognitive distraction driving.

1.3 Driver distraction monitoring system approaches

In the scientific literature there are four main categories according to the measurement signals they used to detect distractions: biological signals, driving signals, driver images and hybrid measures. In this section, the main researches of the four main types of measures will be explored.

1.3.1 Driver biological measures

Biological signals include electroencephalogram (EEG), electrocardiogram (ECG), electrooculogram (EOG), electromyogram (EMG), etc. These signals are collected through electrodes in contact with the skin of the human body and consequently they are intrusive systems [Berka 07, Skinner 07]. Only few works, focusing in cognitive distractions, have been reported in the literature using this kind of approach. Most of them have been analysed in operational environments and not in driving ones. The reason may be that using biological signal to analyse distraction level is too complicated and no obvious pattern can be found. [Berka 07] tried to use EEG data to continuously and unobtrusively monitor the levels of task engagement and mental workload in an operational environment. An inspection on the EEG data using a second-by-second timescale revealed associations between the workload and engagement levels when aligned with specific task events, which provided preliminary evidence that second-by-second classifications reflect parameters of task performance. In [Liu 10], the Kernel Principal Component Analysis (KPCA) algorithm was employed to extract nonlinear features from the complexity parameters of EEG (approximate entropy (ApEn) and Kolmogorov complexity (Kc)) and improved the generalisation performance of a Hidden Markov Models (HMM). The result showed that both complexity parameters decreased significantly as the mental workload increased, and the classification accuracy reached was about 84%.

1.3.2 Driving performance measures

Vehicle signal reflects driver's action, then, measuring vehicle signal driver's state can be characterised in an indirect way. Force on pedals, vehicle velocity changes, steering wheel motion, lateral position or lane changes are normally used in this category. [Farid 06] tried to distinguish between attentive and inattentive driving in car-following situations by analysing the vehicle following distance and steering angle. They built a real time model using Hidden Markov Models with Gaussian Mixtures to infer the intentions of the driver, and this model was able to detect a lane change half a second earlier than conventional approaches. In [Wakita 05], a Gaussian Mixture Model was adopted to identify the driver based on the driving behaviour signals: forces on the pedals and vehicle velocity. [Torkkola 04] adopted the steering wheel position, accelerator pedal position, lane boundaries, and upcoming road curvature to infer driver status. First, the original signals were preprocessed, which yielded a huge set of features. Then, the Random Forest (RF) technique was employed to select the optimal parameters from the derived features. The classifier was also constructed using RF, and the final accuracy reached 80%. In [Ersal 10],

a radial-basis neural-network-based modelling framework was developed to characterise normal driving behaviour. Then, in conjunction with a Support Vector Machine (SVM), it was able to classify normal and distracted driving. Vehicle dynamics and driving performance data such as: vehicle position, velocity, and acceleration, as well as throttle and brake pedal positions were adopted to model normal driving. The average and standard deviations of the residuals (the differences between the actual and model-predicted driver actions) were chosen as the inputs for the SVM. The results showed that the accuracy varied between individuals.

The advantage of these approaches is that the signal is meaningful and the signal acquisition is quite easy. This is the reason because the few commercial systems existing nowadays use this technique [Volvo 10, Mercedes-B. 08]. However, they are subject to several limitations such as vehicle type, driver experience, geometric characteristics, condition of the road, etc. Then, these procedures require a considerable amount of time to analyse user behaviours and therefore, they do not work with the so called micro-sleeps -when a drowsy driver falls asleep for a few seconds on a very straight road section without changing the vehicle signals.

1.3.3 Driver visual measures

Approaches based on image processing are effective because of the occurrence of distraction are reflected through the driver's face appearance and head/eyes activity. Different kinds of cameras and analysis algorithms have been employed in this approach. We group them according to the camera they adopted, including visible spectrum monochrome cameras, IR cameras or stereo cameras.

1. *Methods based on visible spectrum camera*

   The simplest hardware setup is a visible spectrum image acquisition system, but the processing algorithm is relatively complicated, because the problem of face and eyes segmentation could not be avoided. In [Rongben 04] skin colour information is used to segment the face region. This is based on computationally expensive initialisations and is not robust to different lighting conditions and different skin colours. In [Brandt 04] face region is detected with a boosted cascade of Haar-like features, and eyes are extracted by assuming they are the darkest regions in the face, then eye blinks are measured by analysing the optical flow of eye region. But it needed 5s for processing a single image. In [Sun 07] face is detected using adaptive boosting, eyes are located using a template matching method, and gaze is estimated by combining Hough transform and gradient direction. Eye activities do not only contain fatigue information but also distraction information. In [Su 06] a simple approach is proposed to detect driving distraction. After facial area is segmented they perform p-tile algorithm and k-means clustering to locate the eyes. There is a fact that different facial orientations correspond to different types of triangle consist of the locations of the two eyes and the centre of the face. They cluster facial orientation into five clusters: frontal, left, right, up, and down. After facial orientation is obtained, a threshold is made to determine if distraction occurs. Besides analysis of eye activities, mouth activities are also analysed in some researches [Rongben 04, Fan 07, Vural 07] to estimate driving inattention level. In [Rongben 04] they use BP ANN to estimate 3 mouth states from lip features: normal, yawning and talk. [Vural 07] uses a Linear Discriminant Analysis (LDA) to classify the mouth into two states: normal and yawning.

Facial expressions also indicate the presence of distraction. In [Rongben 04], the authors use Facial Action Coding System (FACS) to code facial expression. They employ machine learning to discover what facial configurations are predictors of distraction. This system claims to be able to predict sleep and crash episodes with 96% accuracy within subjects and above 90% accuracy across subjects. But this system operates at only 6 frames per second on a Mac G5 dual processor with 2.5GHz.

A commercial eyetracker is also employed in some research. [Blaschke 09] uses an off-the-shelf eye-tracker to get head pose and eye gaze signal and it models the visual distraction level as a time dependency of the visual focus, with the assumption that the visual distraction level is nonlinear: visual distraction increases with time (the driver looks away from the road scene) but decreases nearly instantaneously (the driver refocuses on the road scene). Based on the pose/eye signals they established their algorithm for visual distraction detection: first, there is a distraction calculation to compute the instantaneous distraction level, second, there is a distraction decision-maker to determine if the current distraction level represents a potentially distracted driver.

2. *Methods based on IR camera*

   Many researches have adopted image acquisition systems based on infrared illumination (IR). The use of IR serves three purposes: firstly, it minimises the impact of different ambient lighting conditions; secondly, it allows producing the bright pupil effect, which makes the eye detection easier; thirdly, since near-infrared is barely visible to the driver, this will minimise any interference with the driver's driving. Because of bright pupil effect the eye can be detected directly, which eliminates the face segmentation and reduces time cost. In [Ji 02] driver's distraction is also detected through face pose estimation. There exists a relationship between face orientation and seven pupil parameters: inter-pupil distance, sizes of left and right pupils, intensities of left and right pupils, and ellipse ratios of left and right pupils. They use eigenspace algorithm to map these seven pupil features to face orientation which is quantised into seven angles: -45, -30, -15, 0, 15, 30 and 45 degree. Besides face orientation, they estimate gaze direction based on information of head movement and relative position between pupil and glint. Like face orientation, gaze direction is quantised into nine zones: left, front, right, up, down, upper left, upper right, lower left, and lower right.

   [Cudalbu 05] uses a similar image acquisition hardware. To estimate the head pose they employ one headband with IR reflective markers, through which they get a 6 DOF head pose with the average error of 0.2 degree. Incorporated with this headband, they use a simplified 3D eyeball model to estimate the gaze orientation with an accuracy varied from 1 degree to 3 degree. [Huang 07] gets the pupil location from a single image: first, getting pupil candidates through Sobel edges, then identifying pupils using SVM with a Gaussian Kernel. In [Jiao 07] a Round Template Two Values Matching algorithm is proposed to locating the bright pupil, which gets an accuracy of 96.4% but consumes 1.01 seconds per frame on a PIII 800MHz computer.

   Some commercial products measuring driver's states are already available on the market, such as SmartEye AntiSleep [Zhou 08, Skinner 07] and SeeingMachines DSS [Carsten 05, Liu 10]. Both of them use two IR illuminators to enhance their robustness to lighting condition, and employ only one camera to give 3D information.

However, they are focusing on detecting fatigue, not distraction, and they are still limited to some well controlled environments.

3. *Methods Based on Stereo Camera*

   Stereo cameras are also employed to estimate driver state. In [Eren 07] two standard web-cameras are employed to make a 3D image acquisition system. They extract face from the disparity map on the assumption that the driver face has smaller depth than background. After face region is extracted they perform embedded HMM to recognise the forehead, eyes, nose, mouth and chin, from which the driving fatigue level can be estimated. The commercial products based on 3D camera technology such as Smart Eye Pro [Skinner 07] and Seeing Machines faceLAB [SeeingMach. 10], can provide measurements of head pose, eyebrow, eye, nose, and mouth.

In conclusion, different kinds of cameras and analysis algorithms have been employed in this approach: methods based on visible spectrum camera, methods based on IR camera, and methods based on stereo camera. Some of them are commercial products as: Smart Eye [Zhou 08], Seeing Machines DSS, Smart Eye Pro and Seeing Machines faceLAB®. However, these commercial products are still limited to some well controlled environments, so there is still a long way to go estimate driver's distraction state. On the other hand, most works existing in the literature were designed for visual distraction detection, less for cognitive and none on auditory and biomechanical distraction detection.

1.3.4 Hybrid measurements

Combining driver physical and driving performance measurements could intuitively increase the inattention detection confidence. On the other hand, road scene analysis and observations of the driver's face would make it possible to estimate what the driver knows, what the driver needs to know, and when the driver should know it. Combining driver gaze information with road scene information offers several potential benefits: context relevant information selection, unnecessary information suppression, and anticipatory information selection.

In [Zhou 08], the standard deviations of eye gaze, head orientation, pupil diameter, and average heart rate (RRI) were combined to improve the accuracy of the driver cognitive distraction detection. The eye and head parameters were obtained using faceLAB, while the RRI data came from ECG. In their work two machine learning techniques, SVM and Adaboost, were implemented under the same conditions. The results showed that the classification performance of Adaboost was slightly better than that of SVM, while the recognition time of Adaboost was approximately 1/26 that of the SVM.

[Sathyanara. 08] tried to detect distraction by combining motion signals from the leg and head with vehicle signals. The motion signals included the 3-axis acceleration of the right leg and 2-axis orientation of the head. The vehicle signals adopted included vehicle speed, braking, acceleration, and steering angle. Then, a group of features were derived from these signals based on the nature of the signals. Next, these derived features were analysed using LDA to reduce the dimension. Then, a K-Nearest Neighbours classifier was trained and verified.

In order to cope with the variability that exists between drivers and manoeuvres [Sathyanara. 10] utilised a Gaussian Mixture Model (GMM)/Universal Background Model (UBM) and likelihood maximisation learning scheme to first identify the driver through an audio signal and then recognise the manoeuvres (right/left turn and lane change) through

CAN-bus signals. Finally, the CAN-bus signals were also used to detect distraction for a particular driver and particular manoeuvre. It was reported that this system could reach an accuracy of 70% for distraction detection. [Doshi 09] fused head orientation detection and a saliency map of the surroundings to determine whether there was a salient object in the driver's view, which gave an indication of whether a driver's head turn was motivated by the goal in his mind or some distracting object/event in the environment. It is known that road geometry influences gaze behaviour and this aspect was taken into account by including road geometry as an additional factor when detecting driver distraction in [Weller 09]. They utilised an analysis of variance (ANOVA) and binary logistic regression to analyse and establish a model for distraction detection based on gaze variables and driving data: fixations (number and duration), scan path, standard deviation of gaze location, speed (minimum, maximum, average and percentage change in speed), lateral acceleration (maximum), and longitudinal deceleration (maximum). The results showed that the road geometry does influence the accuracy of distraction detection based on driving data, but gaze behaviour is mainly influenced by distraction, with little or no influence by road geometry.

[Liang 07] tried to detect the driver cognitive distraction caused by interacting with IVIS in real time by fusing eye movement and driving performance using an SVM. The measured signals included fixation, saccade, smooth pursuit of eye (calculated from raw gaze vector obtained using faceLAB [Skinner 07]), steering wheel angle, lane position, and steering error. These measurement were summarised over various windows to create instances that became the SVM model inputs. After training, the SVM model could detect driver distraction with an average accuracy of 81.1% (sd = 9.05%). [Lee 07b] utilised the same conditions as [Liang 07] but adopted a Bayesian network to detect cognitive distraction, showing that compared to an SVM model, the Dynamic Bayesian Network produced better accuracy.

[Markkula 05] concentrated on processing head/eye and vehicle performance information to estimate both visual and cognitive distractions. The head/eye information derived from stereo cameras included head position, head orientation, gaze orientation, saccade, and blink identification, as well as confidence values. The vehicle performance information included lane position, vehicle speed, etc. Based on the head/eye information, they developed Gaze-World Mapping and Eyes-Off-Road Detection, which could detect momentary visual distraction. Another algorithm, visual time sharing detection, was developed to measure longer term visual distractions. For cognitive distraction, they used 3 indicators to classify the cognitive tasks with an SVM: the standard deviation of gaze angle, standard deviation of head angle, and standard deviation of lane position. However, in the Gaze-World Mapping phase, which mapped gaze and head angles onto actual world targets of visual attention, the road-ahead target was static and determined offline by inspecting the distribution of gaze angles for road-ahead data, and then manually enclosing the distribution in a rectangle.

[Tango 09] proposed a method to derive the distraction level from relevant vehicle and environment data using the Adaptive Neuro-Fuzzy Inference System (ANFIS). Rather than a binary "yes" or "no", they chose Reaction Time as the output to train, validate, and test their ANFIS model. The candidates to be selected as input for the ANFIS included the environment visibility, traffic density, and the standard deviations in speed, steering angle, lateral position, lateral acceleration, and deceleration jerk. After preprocessing, the level of difficulty of operating an IVIS and the standard deviation of steering angle were found to have the highest correlations with the reaction time. Thus, they were selected as the input. However, no accuracy information was provided. [Fletcher 05]

utilised faceLAB to obtain information such as eye gaze direction, eye closure, and blink detection, as well as head position information. In this system, upper and lower bounds were placed on the percentage of time the driver spent observing the road ahead, called the Percentage Road Centre (PRC) [Victor 05]. A percentage that was too high (>90%) could indicate a fatigued state (e.g., vacant staring). A percentage that was too low (<20%) might indicate a distracted state (e.g., tuning radio). Similar to the PRC metric, they analysed driver gaze to detect even shorter periods of driver distraction. They used gaze direction to reset a counter. When the driver looked forward at the road scene, the counter was reset. If the driver's gaze diverged, the counter began timing. When the gaze had been diverted for more than a specified time period, a warning was given. The time period for the permitted distraction was a function of the vehicle velocity. As the speed increased, the permitted time period would decrease, either as the inverse (reflecting time to impact) or the inverse squared (reflecting the stopping distance). They tried to integrate driver's gaze information into other driver assist systems to make the system more acceptable and safer. The framework is shown in figure 1.1. They also spent a significant amount of effort on integrating driver gaze information into lane tracking and sign reading systems. The lane tracking system was used to orient the driver gaze information. A strong correlation was found to exist between the eye gaze direction and the curvature of the road during normal driving [Apostolo 02], with low correlation being a potential indicator of inattention. [Fletcher 05] integrated driver visual information with sign detection to implement a Sign Driver Assist System. This system recognised critical signs in the environment. At the same time, the driver monitoring system verified whether the driver looked in the direction of the sing. If it appeared that the driver was aware of the sign, the information could be made available passively to the driver. In contrast, if it appeared that the driver was unaware of the information, it could be highlighted in some way.

A driver's body posture information is potentially related to driver intent, driver affective state, and driver distraction. [Tran 10] explored the role of 3D driver posture dynamics in relation to other contextual information (e.g., head dynamics, facial features, and vehicle dynamics) for driver assistance. It focused on head pose and upper body posture extraction, but no significant results on driver assistance were found.

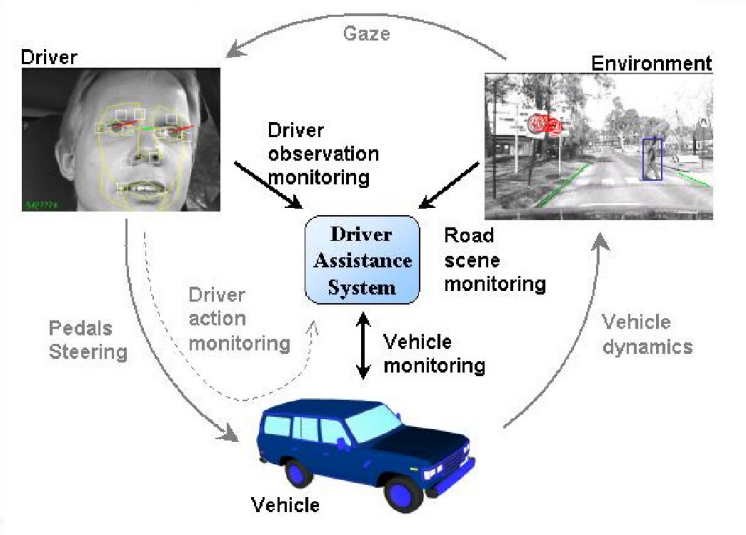


Figure 1.1: Systems integration with gaze information [Fletcher 05].

1.4 General objectives of this thesis

From this review of the State of the art we can conclude that in-vehicle and portable information and entertainment technologies are emerging rapidly, making it increasingly difficult to determine the scope of the potential distraction problem. To date, naturalistic scenarios providing incidence data on distracting activities have been small-scale studied. An effort is needed to study distraction problem using naturalistic situations.

There are different proposals to detect distractions but, to date, they are focusing in some kind of distractions and they do not solve the problem in a general way. Few works has been reported for distraction detection using Driver Biological Measures. This is because using biological signal to analyse distraction level is too complicated and no obvious pattern can be found. Then, individual patterns vary between individuals.

Driver Physical Measures and Driving Performance Measures are the most promising methods in the real driving context, because neither rely on intrusive measurements that might affect the driver. Vehicle signal reflects driver's action and the driver's action is the end stage in the driver's information process. But when the brain's automation function works the driver could drive the vehicle as normal even inattention distraction occurs, which means when distraction occurs the vehicle signal may not reflect the occurrence. Then, this is a driver dependable approach. On the other hand, many of the researches claimed very high detection accuracies, which are true only for their particular hypothesis of distraction definitions. These definitions usually cover a limited region of the whole distraction definition.

On the other hand, since most of the occurrence of distraction can be reflected through the driver's face appearance and head/face activity, the Driver Physical Measures based on Image Processing are effective to detect distraction level. Different kinds of cameras and analysis algorithms have been employed in this approach, and two main stages exist: the first one is producing 2D/3D space information of the driver's head/face organs by image processing, the second one is producing fatigue/distraction level by analysing these 2D/3D information. So far the first stage could produce acceptable result, the research on this stage would focus on improving its robustness, speed cost effectiveness, reducing calibration process , etc. There is large room for the research on the second stage. Same to the other approaches, study on driving distraction is far less active than that on fatigue detection.

As to distraction detection using Image Processing techniques, mouth activity, gaze activity and facial expression are employed. Mouth activity is employed to detect distraction of using cell-phone, while gaze activity is employed for visual and cognitive distraction detection.

Starting 2004, the RobeSafe Research Group[1] from the University of Alcalá, has been working on driver's assistance and safety projects, such as driver drowsiness estimation [Bergasa 06], hazardous conditions [Alcantarilla 08], and driver's face tracking [Nuevo 09]. In the last of these works, Nuevo exposed a novelty method to robustly track the face of the driver, using monocular vision. However, the algorithm developed by Nuevo only works for narrow face rotation angles. To accomplish a correct study on the distraction sequence and behaviour patterns, it is necessary to accurately track the driver's face and gaze over wider rotation angles. This can be achieved with the introduction of a stereo vision system. Other previous works in the group [Bergasa 06] showed that active-illumination system could be applied to face tracking and fatigue detection, based on the

[1] `http://www.robesafe.com/`

red-eye effect on the drivers' pupil.

During the last decades, the number of works related to face tracking and pose estimation has increased, making of this aim a very active research field now a days [Yang 02a]. However, this is still an open problem. Little publications refer to driver distraction pattern and behaviour during real driving. There are only some few commercial companies offering their products for face pose estimation [SeeingMach. 10, SmartEye 09]. However, no technical information on their algorithms has been published. Only in recent years methods tested in real driving conditions have appeared, with some of them using video sequences that feature drivers in a vehicle. From this, two closely related problems can be considered. First, to develop pose estimation and tracking techniques able to work properly on drivers' faces in real driving situations, and second, to infer distraction level by analysing gaze estimation.

This thesis presents an unobtrusive driver's distraction system based on image processing approach. Driver's gaze direction is obtained from a stereo camera and focalisation of the gaze in the scene over the time is analysed in order to estimate driver distraction. Different distraction activities will be inferred in a naturalistic simulator and a study of the incidence of these distracting activities in the driver will be carried out.

1.5 System requirements

The work of this thesis is focused on generating the necessary tools to estimate drivers' face pose and gaze on a naturalistic simulator so that some experts in psychology can analyse driver distractions. This goal implies that the result must be an out-of-the-box component, easy to work with. Since it is intended for a simulator, an initial installation and calibration process by qualified personnel is allowed, but after that the system must be easily operated by non technical personnel. Consequently, operation after installation must be fully automatic, without requiring any specific driver calibration.

In addition, simulators usually present low light conditions to increase the user immersion feeling. The developed method must work on low light conditions and with a good precision. We propose to use cameras with high sensibility and an external IR light source to obtain good face appearance without annoying the driver. However, it's well known that IR illumination can produce visual fatigue. To minimise this effect our proposal is to synchronise an active illumination system with capturing using a relatively low level intensity. Finally, the system must be able to operate in presence of wide head turns and must be robust to partial occlusions, different users —with and without glasses— and slight illumination changes.

As conclusion, the requisites of the system to be developed are the following:

- **Two cameras**, of visible spectrum with high sensitivity to near-IR, able to provide at least 30 frames per second.
- **Synchronisation hardware** to drive both cameras and IR illumination.
- **Real-time** operation, at the frame rate of the camera.
- **Automatic operation with any user**.
- **Night-time** operation.
- **Automatic face model** generation.

- **Robust operation** in presence of **wide head turns** and movements.
- **Robust operation** in presence of **partial occlusions**, either by the driver's hands or any other external element.
- **Automatic and fast localisation of the face** after tracking losses.
- **Analysis of the gaze focalisation** in the scene.
- **Distraction inference** from the gaze analysis.

1.6 Document structure

This document is divided in chapters, of which this introduction is the first one.

Chapter 2 reviews the State of the art on face pose estimation techniques, as well as 3D reconstruction and tracking methods.

Chapter 3 introduces the general architecture of the method proposed in this thesis for face pose estimation. The details of the proposal are explained in chapters 4 and 5, which describe how a 3D face model is created and the methodology to estimate the pose based on that model, respectively. The gaze estimation calculation, as a composing vector of face and eye direction is presented in chapter 6.

Chapter 7 explains the testing methodology, the video sequences used for evaluation and ground-truth, and the results obtained by the method for face pose estimation, gaze direction and distraction parameters inference. Finally, chapter 8 contains the conclusions, summarises the main contributions of this work, and presents future research lines that could help to improve performance and results.

Chapter 2

State of the Art

Face pose estimation has been a very active researching subject for more than two decades. During this period, the techniques have evolved together with the increasing computational resources of modern computers. Along with this evolution, the objectives of face estimation systems have also became more enterprising. The first proposed works only aimed to detect a few predefined poses, just enough to allow a coarse pose estimation. Those systems would enabled a machine to discriminate the interlocutors of a conversation inside a room with a controlled light environment.

Nowadays, as the basic objective of getting a fine pose estimation is being met, new requisites would be desired depending on specific applications. The technique and technology have evolved, but still there is room for improvement. New systems are much more ambitious. Some modern pose estimators have errors as low as 2°, but new applications may require the systems to work in real-time, or changing or low light condition, user independently or other similar very challenging requisites. This new challenges must be addressed by more intelligent face pose estimation systems which are yet to come. Often, the pose estimation algorithm is just a necessary previous step for a gaze estimation. It is actually the gaze what gives the real information of the point of attention of a subject. Having an accurate gaze estimation requires a very precise face pose estimation.

Automotive industry provides two examples of very challenging scenarios: driver's behavioural study and driver's inattention monitoring. The former, requires a very accurate gaze pose estimation, the collection of time statistics, good level of system integration with other IVIS (on-board and off-board applications), and often low light conditions to work inside driving simulators. Inattention monitoring is probably the worst possible lighting scenario. A user can be driving in a sunny morning, late at night, or throw a forest that projects impossible shadows over the driver's face. In addition, inattention monitoring requires as well requisites such as accuracy and good integration. Added to this, a consumer on-board application would require completely independence of the user, no matter age, gender or raze, no calibration step, and fast initialisation.

The work presented in this thesis focused on providing a system for distraction analysis and driver's behavioural study inside a simulator. To date, it is not possible to find a system which endorses all the mentioned features, and consequently which could be eligible for an on-board distraction monitoring system, which could be commanded by a group of psychologists and other non-technical qualified personal. Providing such a system is the main goal of this thesis.

This chapter presents a review of the most notable methods used for face pose estimation during the last two decades. Several authors have published extended surveys of the literature [Murphy-Ch. 09, Hansen 10]. Main referenced works will be presented below.

For a deeper explanation of the face pose estimation techniques we refer the readers to that surveys. The literature concerned with 3D pose estimation of a generic object is very large and a survey is beyond the scope of this thesis. A review of some existing databases is also presented for reference, since many of the works discusses below base their results in any of these databases.

2.1 Face pose estimation methods

The pose is generally described as a rotation and a translation transformation. This transformation defines the pose with respect to an initially predefined position and orientation of the observer.

Many publications refer to that problem as head pose estimation, while many others name it as face pose estimation. More precisely, the term *face pose* is most commonly used for every technique that involves detection or recognition of a human face, while the term *head pose* is more typically used when only pose estimation is needed. However, the face and the head pose are joined together in all sense, since it is assumed that the human head can be modelled as a disembodied rigid object. Consequently, both terms, face pose or head pose, are interchangeable for any practical purpose. In this thesis we will use the term of face pose, which actually means the same of head pose.

First, it is important to analyse the motion range of a human head in order to know the range that a vision system must cover. Restricting the motion to only the neck, the range of head motion for an average adult encompasses a sagittal flexion and extension (i.e., forward to backward movement of the neck) from -60.42° to 69.6°, a frontal lateral bending (i.e., right to left bending of the neck) from -40.9° to 36.3°, and a horizontal axial rotation (i.e., right to left rotation of the head) from -79.8° to 75.3° [Ferrario 02]. This ranges are represented on figure 2.1. However, the head can also be affected by other rotations, and specially by translations directed from the chest, so the coverage area of the system must be wider. Indeed, different individuals may perform the same movements with peculiar modalities and diverse contributions from dorsal spine and thoracic girdle. The total range of movement that the system must be capable of depends on the user, the application and the environment where the user is being monitored.

It is possible to classify the different approaches to face pose estimation attending to many different criteria. But the important aspect to which this State of the art is

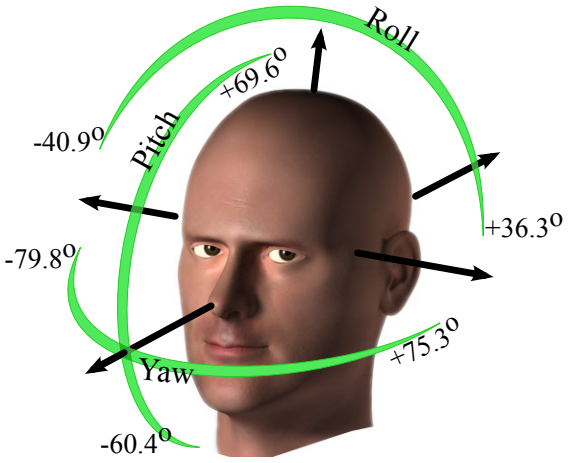


Figure 2.1: Head rotation angles

focused is the output range and the technical approach used. To start off, systems can be differentiated by its output range in two groups:

1. **Coarse head pose estimation** systems, for which output is within a discrete range of positions called *poses*. The output pose is one of a finite set of *a priori* defined poses. In early years many pose estimation systems of this type were designed. For instance, they were capable of distinguishing interlocutors in a conversation. These systems are not adequate for gaze estimation, since they are not able to represent the nature of the continuously moving gaze and consequently do not meet the conditions requested in System requirements, section 1.5. To provide a comprehensive overview of the State of the art, a sort review of those systems will be presented, although they are out of the interest of this thesis.

2. **Fine head pose estimation** systems. In this case, the output pose is a continuous and fine output range, i.e., the estimated pose is an analog non-discrete solution. To this group belong most of the solutions which give better accuracy, and they are well suitable as a previous step for gaze estimation. This is the group in which the work of this thesis can be included.

Attending to technical aspects, Murphy-Chutorian and Trivedi [Murphy-Ch. 09] classified head pose estimation systems in eight categories. This arrangement is done by the fundamental technical approach behind the implementation of each system. Following are the most representative works for each of the different approaches.

2.1.1 Appearance template methods

These methods try to locate the head by comparing the face patch image to patterns of labelled poses learnt in advance. This technique has the advantage of its simplicity and ability to process low resolution images. However, it can only produce discrete predefined poses, one for each of the templates used. Some characteristic examples include the use of normalised cross-correlation at multiple image resolutions [Beymer 94] and mean squared error (MSE) over a sliding window [Niyogi 96]. The common problem to this approach is that two different subject with two different poses can have very similar appearance, and thus the system outputs the same pose estimation for both, one of them being obviously wrong. In general, pose estimation is inaccurate for many users. Experimenting with some image transformations, such as a Laplacian-of-Gaussian filter, can emphasise some of the more common facial features, improving the algorithm accuracy [Gonzalez 02]. This technique usually requires a previous manual and tedious image labelling to identify templates with output poses. The output is often a coarse head pose estimation.

A typical human head depicts a cylindrical shaped structure, so the face can be approached to a vertical oriented cylinder. Based on this, many authors apply cylindrical texture mapping to deal with appearance variation of the face due to rotation [Lin 09]. This technique, depicted in figure 2.2, has been successfully applied for this and many other different approaches. However, it requires an accurate head detection and localisation step in order to be able to calculate the correct cylinder projection which best fits the face.

2.1.2 Detector arrays

These methods train a discriminant classifier in advance, such as an Artificial Neural Network [Rowley 98], and process the input image with it. This technique is similar to

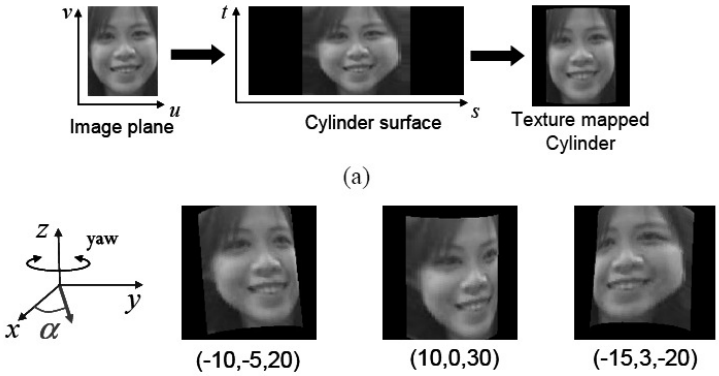


Figure 2.2: Examples of the 3D texture mapped cylinder rotated and rendered with OpenGL libraries. The notation under graph denotes (pitch, roll, yaw). [Lin 09]

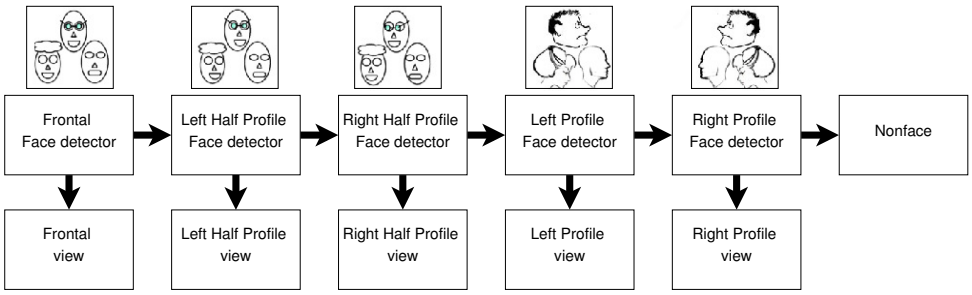


Figure 2.3: Detector array classifier for local face detection [Zhang. 07]

the previous one, except that instead of using image matching with all the cases in the training database to find the most likely pose, it uses a binary classifier trained with all the faces sharing a same pose withing the database. It allows to train the detector to distinguish not only the different head poses, but also the presence or not of a face, and so the face localisation and pose estimation process can be achieved in one step. An recent work on this technique was proposed by Zhang *et al.* [Zhang. 07]. In this paper, a set of 5 multi-view face FloatBoost classifiers are applied to estimate head pose in seminar room scenario. Figure 2.3 depicts the classifier array used in their work. Naive Bayesian is used to fuse estimation of head pose from four camera views, and Hidden Markov Model is used to model the temporal change of the pose in the video sequence. This system is able to work on low resolution images and it can be used to detect coarse poses for many interlocutors inside a monitored room. Other typical approaches is using Support Vector Machine (SVM) as the classifier [Li 02, Seo 04, Li 04]. As in the previous group, the main disadvantages of this approach is that only coarse poses can be accurately obtained and a tedious training process is needed.

2.1.3 Nonlinear regression methods

The basic idea of nonlinear regression methods is the same as that of a linear regression, namely to relate the pose response i.e., one or more angle of rotation, to a vector space formed by a set of input images, which works as the predictor variable. This method can provide a reliable output for any new input image if an appropriate input set of labelled data is provided. This input set can be trained online, notably easing the process of training. Success using nonlinear regression has been demonstrated using Support Vector

Machine (SVM) [Li 04], Support Vector Regression (SVR) [Murphy-Ch. 07] or Neuronal Networks [Seemann 04]. Figure 2.4 depicts an example using this approach. Murphy-Chutorian's *et al.* approach overcomes the difficulties that arise with varying lighting conditions in a moving car by means of Localised Gradient Orientation Histograms to tolerate deviations caused by scale, position, rotation, and lighting. Using this representation, they reduce the dimensionality of the input data, providing a stable input to a SVR for robust head pose estimation in two degrees-of-freedom [Murphy-Ch. 07]. In general, the accuracy of the nonlinear regression methods depends heavily on the results of the previous head localisation process. Despite the output estimated pose being continuous, the regression curve is made up with a finite set of poses. If a given input pose happens to be in between two of the poses used to estimate the regression curve, it is not clear how well the regresor can relate this input pose to the response curve. This implies that accurate estimation can only be obtained for discrete poses. Moreover, the estimation speed is related to the dimensionality of the regression tool used. In general, time increases nearly linearly with the increase of dimension, and a poor real-time performance is observed owing to the large number of dimensions needed for an accurate approach. Chutorian's system, for example, runs at approximately 5 frames-per-second[1], limited primarily by the time required to process the Adaboost cascades.

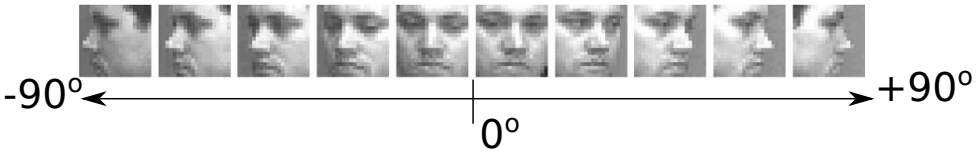


Figure 2.4: Nonlinear regression from face image to pose estimation [Li 04]

2.1.4 Manifold embedding methods

A recent tend in head pose estimation research has been the use of manifold learning techniques to capture the underlying geometry of the images. Face images with varying pose angles can be considered to be lying on a smooth low-dimensional manifold of the higher dimensional image space [Balasubrama. 09]. Some approaches tested using this method include Principal Component Analysis (PCA) [Artac 02], and its variation KPCA, which uses nonlinear kernels [Wu 08], Locally Linear Embedding (LLE) [Balasubrama. 07] or Locally embedded analysis (LEA) [Fu 06], among others. Remarkable is the work proposed in [Fu 06], with an yaw error lower than 2°. They minimise the local reprojection error by constructing the manifold graph applying a supervised LLE approach, formally LEA. The head pose is finally estimated by a K-nearest neighbour classification. The manifold graph they build can be observed in figure 2.5. Although they obtain a very low error, their algorithm is only applied in one degree-of-freedom (DOF): yaw rotation.

With real-world face images, manifold learning techniques often fail because of their reliance on a geometric structure, which is often distorted due to many variants. Embedding techniques trend to include in the different training data subjects' identity into the manifold space, which lead to a pose estimation not totally independent on the subject. The same as the previous methods, this technique requires training of a database. The linear versions, such as PCA, for instance, can only handle linear spaces, which do not represent the problem of a 3DOF rotation very well. Its nonlinear variation, KPCA, on

[1]Hardware platform is not specified

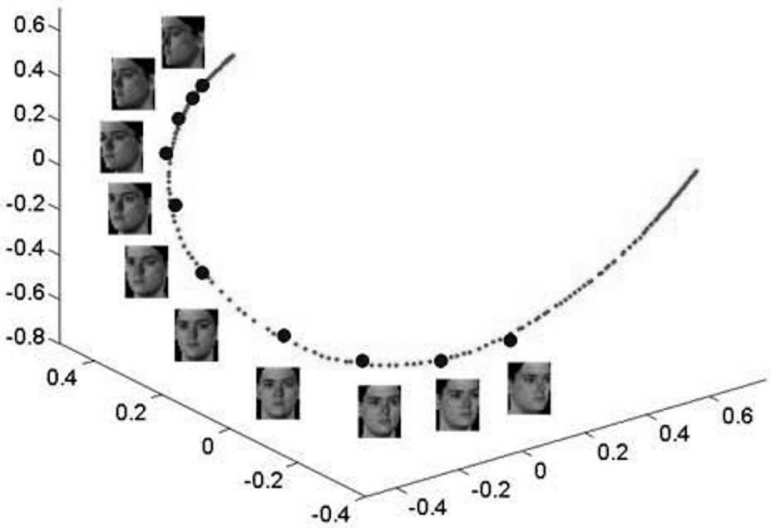


Figure 2.5: Linear embedding and subspace projection of 181 view rotating images of a face, constructed using a supervised variation of LLE: LEA [Fu 06]. The curve represent the structure of the manifold embedding in the reduced, low-dimensional space

the other hand, can not be applied using matrix multiplication, what makes it slower.

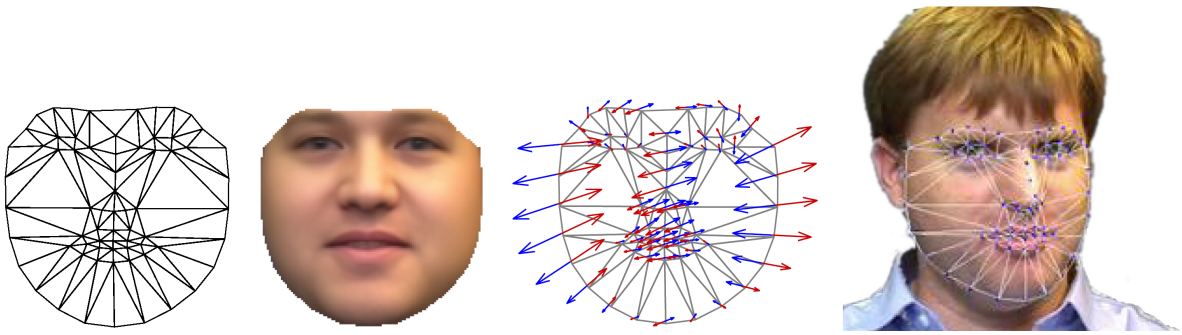
2.1.5 Flexible models

These techniques are based on fitting a flexible model to the structure of the face. Active Shape Models (ASM) [Cootes 02, Cootes 05] are deformable models that have been demonstrated with good results. The shape is used to search faces and estimate its parameters, i.e., the pose. ASM has been later extended by other researchers to include robust fitting functions [Rogers 02] and stacked models [Milborrow 08]. Advances in late years have increased their robustness and precision to remarkable levels [Milborrow 08]. In most works, models are 2D. Extensions of ASM that include modelling of texture have been presented, of which Active Appearance Models (AAM) [Cootes 01a] are arguably the best known. Few tests employed images recorded on cars [Baker 04b], but these sequences were short and did not contain challenging scenarios. Both ASM and AAM are linear models, and have difficulties in tracking faces under partial occlusions by external objects, or self-occlusions during head turns. [Xiao 04] use AAM with a 3D shape model, such as the one depicted in figure 2.6, which improves pose estimation accuracy, but still it needs most of the face visible, and deal badly with occlusions and wide head turns. Moreover, training a model such as AAM or ASM is a time consuming task, as it usually involves introducing landmarks by hand in hundreds or thousands of images.

In this group we can also include the non-rigid structure-from-motion approaches. These methods creates an online model of the face, assuming from the beginning that the face is not rigid, and its structure can change during the execution, after the initialisation process. This is somehow very similar to an ASM, except that ASM uses an *a priori* model definition, unless deformable, and the former starts creating it from scratch. Paladini *et al.* [Paladini 10] use an incremental non-rigid Structure from Motion (nr-SfM) model by adding new deformations incrementally when the current can not model the face well enough. They consider the image reprojection error to decide when to update, and use bundle adjustment to estimate the new mode to be added. An advantage of this approach is that it does not rely on prior knowledge of the model. However, it is not clear if this

method could work on real time, due the time consuming task of creating new modes.

In addition, the non-rigid model approach is in any case based on an underlying initial rigid model, to which the deformable modes are added. Errors on the initial model are carried out during the whole execution. Another still challenging task for non-rigid models is how to differentiate tracking errors from deformations.

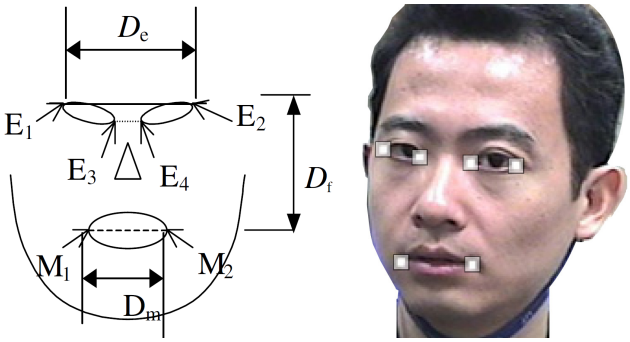


(a) Face flexible shape model (b) Face Appearance model (c) 2D deformations vectors applied to the model (d) 3D deformable model fitted to a face for pose estimation

Figure 2.6: A face flexible AAM model [Xiao 04]

2.1.6 Geometric methods

Tracking a face using the most salient parts of it has a greater chance of success than relaying on any other less defined parts. Geometric methods exploit this to get a relatively good accuracy and robustness. One of the most common approaches is to use templates to track these prominent face parts. A typical set of landmarks can be the nose, mouth and eye corners. [Wang 07] presented a method for computing head pose from a single image by using projective invariance of the point at infinity. This last approach assumes full perspective projection camera model. An analytic solution was derived and the pose is determined uniquely when the ratio of the length of the eye-line segment to the length of the mouth-line segment is known (See figure 2.7). This technique achieves very good accuracy, with an error as low as 1.67° on small yaw rotations. However, estimation relays on very little key face features, and so deal very badly with occlusions and noise.



(a) Geometric model, based on geometric relations of face landmarks (b) Pose estimation using the six point geometric model

Figure 2.7: Geometric model based on six face landmarks: Two outer eye points, E_1 and E_2, two inner eye points, E_3 and E_4, and two mouth corners, M_1 and M_2 [Wang 07]

A limitation of these approaches is that they require that all the facial features, including the eye and mouth corners, are visible in all frames, restricting rotations to no more than 30°. For instance, the work presented by Wang only covers rotations of bellow 30°. The allure of this approaches is that this is a simple and incredible fast algorithm compared to any other, since it only requires finding a small set of prominent face features and some geometrical calculations. On the other hand, any small error on the localisation of any of the few features used will generate a big estimation error.

These methods are probably the weakest in terms of robustness and reliability. The pose is based on the initial geometric assumption of the model, which usually is acquired in the first frame. Any small error due to initialisation, landmark localisation, user gestures, occlusions or rotations lead to a high pose estimation error.

2.1.7 Tracking methods

These methods estimate the pose parameters by using feature correspondences between two frames. This is somehow similar to the geometric methods explained above, except that none of the former considers the problem of large motion, specially for yaw rotations. Tracking techniques handle this problem following the head rotations along its whole range, by individually tracking selected features of the face.

Some of the methods detect new features frame to frame and calculate the rotation and translation variations. In this case, the pose would be given by the accumulation of the step by step pose variations frame to frame. This approach is very sensitive to local errors, since the error of a frame estimation accumulates to the next. This is typically a major problem to solve in a *Simultaneous Localisation And Mapping* (SLAM) system. To avoid this accumulative error, other typical approach is based on tracking the same set of features frame to frame since the beginning, and estimates the pose always referencing to the first frame. This is possible since the total head movement is much smaller than that showed on SLAM problems. [Sheerman-C. 09], for instance, presented a facial feature tracker that works without *a priori* knowledge of the appearance of the face, using Lukas-Kanade (LK) [Tomasi 91] tracking points, and an online learning scheme to update the tracking points templates. Similar techniques are widely applied by many authors since they have some advantages. First, a model can be constructed with the features, allowing for a further correction step based on a higher level, model-based algorithm. Second, as mentioned above, they do not present accumulative error because of the frame to frame pose estimation. On the other hand, they suffer from template drifting. As the faces rotates it is necessary to update the templates used for tracking. Smalls error in updating accumulate, so the real features being tracked may differ from the original ones after a while.

[Zhao 07] tracked face pose frame to frame by means of Scale Invariant Feature Transform (SIFT) based registration algorithm. Salient SIFT features are first detected and tracked between two images, and then the 3D points corresponding to these features are obtained from a stereo camera. With these 3D points, a registration algorithm in a RANSAC framework is employed to reject the outliers and estimate the head pose using full perspective projection. Performance evaluation showed an accurate pose recovery (3° RMS) when the head has large translations and rotations in the range of ±45°. One major drawback of this tracking method is that correct correspondences can only be obtained when the pose variation between two frames is small enough. This is actually one of the weakness of the tracking methods. In general, features must be tracked frame to frame. If the face experiments a big appearance variation from one frame to another, the feature

tracker may be lost, and the algorithm must be restarted from a predefined position.

Another commonly used approach of these methods is the model tracking. Unlike [Zhao 07], which follows different and new features frame to frame, a model approach generates global features which will be followed across frames during the whole execution time. These features make a model which can be fitted after individual feature locations have been found. In this sense, quite different formations are possible. [La Cascia 00] used a manually initialised cylindrical model and recursive least squares optimisation to track the head. Their idea consists on composing and warping the face images from different view points to a cylindrical surface model to approximate the head, accounting in this way for self-occlusions and to approximate head shape. Then, they use image registration in the texture map to fit the incoming data. [An 08] built an ellipsoidal texture model of the head and determined pose by matching projections to live images. This avoids dependency on high-resolution images while tracking the full range of orientations, but nevertheless requires initialisation for each subject and static illumination. To solve the illumination problems, [Wu 00] used a similarity metric by means of an ellipsoidal model of points, where each point maintains probability density functions of local image edge density based on training images. [Xiao 02] also used a cylindrical head model and recovered motion using perspective projection. They included the more robust image gradient information instead of edges. The face tracking is implemented by means of a texture based template fitted to the cylindrical model, as in [La Cascia 00], with the improvement of dynamically updating the template to meet the appearance variations due to back-projection and illumination. [Murphy-Ch. 08] use a 3D face model to track the face with a particle filter and a dual state movement model: a dynamic movement model when the face is detected to be moving, and a static one otherwise.

A 3D model with stereo tracking is also possible. [Jiménez 09] use an stereo rig to automatically create it out of 3D features from the face. This model allows the detection of tracking errors that otherwise, with a 2D one, would not be perceptible. They extend the model during execution including new features from previously concealed parts of the face, as new features are exposed to camera. Figure 2.8 shows a functional schema of this system.

Many of the works using tracking and a face model assume it is rigid, which do not completely satisfy the nature of a human face. Deformations produced by gestures can lead to pose errors. The tracking methods approach typically suffer from another common error: the pose offset at initialisation. The initial pose estimation offset is a consequence of the position held by the user at the initialisation frame. The pose estimation on every frame is referenced to that of the first, so if the pose at the initial frame has an offset

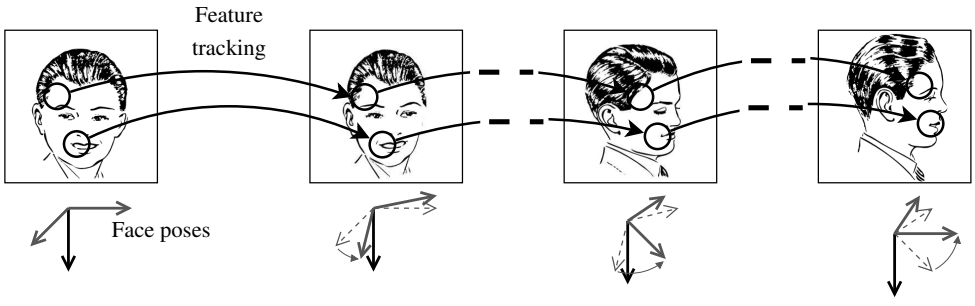


Figure 2.8: Stereo tracking method to estimate face pose based on a incremental 3D model [Jiménez 09]

error, this offset is carried over the whole algorithm execution.

2.1.8 Hybrid methods

Other approaches combine different methods from those mentioned above. Those can be considered as hybrid methods. Mixing together different approaches can help to overcome the problems of any single method. Many of these limitations have been previously discussed. The most typical combination is a coarse pose estimation method to provide a robust initialisation, followed by a tracking loop, which allows for better accuracy than any other. The coarse estimator can be any of PCA or other embedding methods, geometric methods or appearance templates, among others. This coarse estimator is responsible for the correct initialisation and for pose checking over time to correct the accumulative error on the pose estimation.

[Morency 03] presented a method based on an *a priori* model of intensity and depth viewbased eigenspaces, built across multiple views of the head. Given an initial frame with unknown pose, they reconstruct a prior model for all views represented in the eigenspaces. To track more robustly over time, they added an extension by integrating the prior model approach with an adaptive differential tracker. They demonstrated very good accuracy on face pose tracking using stereo cameras. Their approach is user independent and the prior model can be automatically initialised from any view point of the view-based eigenspaces.

[Murphy-Ch. 08] use a classical tracking method based in a 3D model to perform the fine face pose estimation, while a nonlinear regression detector is run periodically to check the consistency of the estimation.

More sophisticated hybrid methods are also possible. In [Krinidis 09], a tracking technique that utilises a 3-D deformable surface model to approximate the facial image intensity is used to track the face in the video sequence. Pose estimation is then achieved by means of a Basic Radial Function neuronal network. This method produces very good results for relative frontal faces. However, the deformable intensity model can not adapt in case of partial occlusions or fast head movements. Moreover, the RBF makes the algorithm very slow, being able to run at 4 fps[2].

2.2 Existing video databases

With the purpose of providing a common framework to compare the different publications and methods for face pose estimation, several video datasets have been created and are commonly used in the literature. Some of these datasets were created for international workshops held in last few years. Others have been created by institutions to provide ground-truth data for researchers.

[Murphy-Ch. 09] described the most common video databases for head pose estimation evaluation. This section introduces a short description of the most representative ones, which have been used for performance evaluation on different works mentioned in this chapter.

A. **CHIL-CLEAR06/07:** [Mostefa 07] This video dataset consists of 15 videos shot from different cameras, inside seminar rooms and lectures, where some few people are interacting. The ground-truth was captured using magnetic sensors, but the database also contains manual annotations on interlocutors every second. The goal for the head

[2]No code optimisation, running a Pentium 4, 3GHz, and 1.5GB of RAM

estimation systems is to continuously track the presenter's horizontal viewing direction in the global room coordinate frame. The dataset is aimed to human interaction applications, such as interlocutor detector, areas of attention, etc.

The images provided by the wide-angle field-of-view camera vary in resolution and speed, from 640 × 480 to 1024 × 768, and from 15 to 30 fps. The cameras are located on the four corners of the room, looking inwards.

The video dataset also provides audio records, two more videos from a fish eye camera and a pan-tilt-zoom (PTZ) camera[3], and annotations on audio, tracking persons and face detection and identification. This data base is available online on the CLEAR web site at `http://www.clear-evaluation.org/?The_Evaluation:Sample_data`. The face resolution is very low, and consequently the dataset is not adequate for this thesis.

B. **BU Face Tracking**: La Cascia *et al.* collected a video dataset for the evaluation of their work [La Cascia 00]. They collected two sets of video sequences, one under uniform illumination conditions and another under time varying illumination. The former contains five different subjects, and the second three. For each subject, 9 sequences of 7 seconds at 30 fps were recorded. The video size is 320×240. All of them depict free head motion in rotation and translation. The ground-truth was captured by a 3D magnetic tracker attached to the head which provides an accuracy of ±2.5 mm in pose and ±0.5° in rotation.

   The dataset is available online at `http://www.cs.bu.edu/groups/ivc/HeadTracking/`. However, the rotation range of the face in the videos is very small.

C. **CVRR LISAP-14**: [Murphy-Ch. 08] The CVRR LISAP-14 dataset contains 14 video sequences of an automobile driver while driving in daytime and nighttime lighting conditions (eight sequences during the day, four sequences at night with near-IR illumination). Each sequence is approximately 8-minutes in length, and includes head position and orientation as recorded by an optical motion capture system. The videos are 640x480 pixel grayscale, at a frame rate of 30fps. More information may be found on request at `http://cvrr.ucsd.edu`.

D. **IDIAP Head Pose**: [Ba 04] This data set was created in 2003. It is part of a series of datasets containing head pose, speech and people interacting. The main objective of the dataset is to provide a common framework to researchers for rigorous algorithms comparison.

   It provides to different types of videos, from which the seconds is focuses on head tracking on Augmented Multi-party Interaction (AMI). It consists on videos recorded on meeting rooms, where at least two people are always in front of the camera. This set contains 8 videos of a duration of 1 minute each, taken from a single camera. Two people are sited in from of the cameras, and perform pitch and yaw rotations of up to ±90° in both directions.

   The ground-truth contains pitch, yaw and head position information, and was captured with magnetic sensors. This video dataset can be downloaded from the Idiap web page, `https://www.idiap.ch/dataset`. However, no roll rotation is provided.

Note that none of the datasets described above provides stereo images, and consequently are not suitable for performance evaluation in this thesis. Moreover, most of

[3]A PTZ camera is a motorised unit, usually controlled by an Ethernet connection, capable of zooming and rotating its field of view

them capture low resolution images, in a good illuminated environment. These conditions differ from those present in a naturalistic simulator, hence these datasets are not suitable to evaluate the algorithms developed in this thesis.

2.3 Discussion

The huge number of works described above shows that there is an intensive effort by researches worldwide on this area, who have developed a wide range of approaches. Despite this, it is hard to find works focused in the study of drivers distractions, which is the intended application of this thesis, as stated in the introduction. Tables 2.1 and 2.2 resume the most significant works, its main characteristics and their limitations.

Many of the works of the State of the art do not provide a fine pose estimation output. Those systems are not suitable for the scope of this thesis, since they do not allow for an appropriate gaze estimation for a driver distraction monitoring application. The methods which produce a coarse output have the advantage that do not rely on tracking systems, minimising the possibility of tracking losses. Those would be very suitable for human-human or machine-human applications, such as identifying the interlocutor of a conversation, or switching panels of an interface. However, their lower accuracy makes those approaches unfeasible for gaze estimation. Generally, this is achieved by composing the face pose estimation with the eye directions [Hansen 10]. Consequently, if the pose estimation is no precise enough, the gaze estimation will be inaccurate as well. From the literature described above, template methods, detectors arrays and nonlinear regression methods must be discarded for this reason. In most cases, the error for the listed methods is higher than 10° in the pose estimation [Brown 02]. Only [Grest 09] show an error close to 5°, with a maximum range of ±20° in yaw and pitch. Another important fact which should be pointed out about these methods is that there are very little published works during the last 5 years. Appearance methods were the most active research in the 90's. The most recent publication using detectors arrays date to 2006, and present an important lack of accuracy.

More recently, one of the most used methods are manifold embedding techniques, such as PCA. It became very popular, specially in hybrid architectures, used in conjunction with other approaches such as flexible models [Cootes 95, Lanitis 97] or more recently with tracking methods [Huang 04, Tu 06]. These dimensionality reduction methods require training and often manual labelling. Consequently, these are usually not user independent. The more users included in the training data set, the more user independent the system might be, but the less accurate the output. Manifold embedding techniques have their main disadvantage in the inability to separate identity and pose estimation, as the number of users in the training dataset grows. This means that the pose estimation can vary for different users [Balasubrama. 07], if the training database is big enough. Moreover, PCA is a linear approach, and consequently is not well suited for the nonlinear problem of a 3D rotation appearance variations. Some authors applied the Kernel-PCA [Wu 08] variation to address this non linearity. PCA and KPCA are also expensive techniques in terms of training requirements. On the other hand, these methods are a good option for low resolution images, where the little texture information available is well exploited by the dimensionality reduction provided by the embedding.

The non-rigid models also present some problems. The process of calculating new modes for a deformable model is slow. If many modes are allowed, there is a chance that tracking errors of rotations are interpreted as deformation. But as the number of allowed

Approach	Error [Yaw-Pitch-Roll]*	Range [Yaw-Pitch-Roll]*	Implementation details	Calibration and initialisation	RT
Appearance template methods					
Relevance Vector Machine [Lin 09]	4.1° 2.3° 2.4°	±80° ±25° ±10°	Landmark localisation using Neural networks+constrains model+Relevance Vector Machine	manual initialisation and training	15 fps
Detector arrays					
SVM [Li 04]	9° 8° -	±90° ±30° -	Multiviews face detectors + PCA	training required	
Nonlinear regression methods					
Neuronal Networks [Seemann 04]	7.5° 6.7° -	±90° ±90° -	Stereo to get depth	network training†	RT
SVR [Murphy-Ch. 07]	6.40° 5.58°	±30° ±80°	Localised Gradient Orientation Histograms		
Manifold embedding methods					
Biased Manifold [Balasubrama. 09]	1.44° - -	±90° - -	Biased Laplacian Eigenmaps	network training†. manual	
Flexible models					
AAM [Xiao 04]	3.8° 3.2° 1.4°	±75° ±40° ?	AAM + 3D shapes		RT
Non-rigid models [Paladini 10]	?	?	Incremental+ ng-SfM + tracking		
Geometric methods					
Face Features [Wang 07]	1.67° 2.56° 3.54°	±40° ? ?	Probabilistic learning of face model + full perspective projection + vanishing point	Manual feature initialisation. No occlusions	

* yaw/pitch/roll data is displayed in top-down order. "-": the proposed work does not cover this DOF. "?": data is not provided in the publication.

Table 2.1: Comparison of related face pose estimation works

Approach	Error [Yaw-Pitch-Roll]*	Range [Yaw-Pitch-Roll]*	Implementation details	Calibration and initialisation	RT
Tracking methods					
Dynamic Templates + Cylindrical model [Xiao 02]	3.8° 3.2° 1.4°	±75° ±40° ?	Cylindrical model + image gradient + dynamic templates & re-registering	No significant face appearance variation required	
Particle Filters [Oka 05]	2.86° 2.34° 0.87°	±40° ±20° ±10°	Stereo + Adaptive control of diffusion factors	No significant face appearance variation required	RT
SIFT [Zhao 07]	2.44° 2.76° 2.86°	±45° ±45° ±45°	stereo+perspective projection + frame to frame correspondences + RANSAC		RT
Ellipsoidal model	3.9° 4.0° 2.8°	±50° ±10° ±10°	Face texture mapping to the model	Generic face model	
LK + online template learning [Sheerman-C. 09]	4.2° 3.9° 3.1°	±20° ±40° ±30°	re-registering learning + LM + RANSAC	Requires *a priori* defined features patches. Strict face constraints	
Hybrid methods					
Prior model + Tracking [Morency 03]	3.2° 1.6° 1.3°	±30° ±20° ±15°	Intensity and depth viewbased eigenspaces +adaptive differential tracker	training required	
Nonlinear + tracking [Murphy-Ch. 08]	3.39° 4.67° 2.38°	±90° ±45° ±45°	Dual state model + SVR + 3D model-based tracking + Particle filters		RT
3D deformable model [Krinidis 09]	2.2° 2.6° 0.5°	±60° ±30° ±15°	Intensity elastic model + RBF	network training	

† Results for other user not in the training set.

* yaw/pitch/roll data is displayed in top-down order. "-": the proposed work does not cover this DOF. "?": data is not provided in the publication.

Table 2.2: Comparison of related face pose estimation works, continuation

modes is decreases, the system gradually loses it non-rigid capability. [Paladini 10], for instance, saturate the number of modes to 10, what is likely to happen during the first minutes of operation. This avoids increasing execution times, but actually limits the learning process in time.

Much faster algorithms may use flexible models, such *Active Appearance Models*, AAM, or *Active Shape Models*, ASM. Many related works on this line also include an underlying PCA for computing the appearance of face landmarks from training images [Baker 04b], having similar problems than the pure based PCA approaches. Flexible models require extensive manual labelling of various face landmarks. Using an extensive database, those methods are user independent. [Nuevo 10] showed that it is possible to achieve very low computational cost using a patch clustering approach. However, the main disadvantage is that they are not suitable for wide head rotations. Related works, such as [Xiao 04, Gui 06], do not show rotations wider than 45°. In addition, the shape, models, even 3D models, often tend to learn small rotations as deformations, not providing an accurate pose estimation.

The same rotation limitations observed at flexible models is present if geometric methods are used. These methods are extremely fast since they only require tracking a few face landmarks. However, the few reference points they use makes these systems sensible to tracking errors, gestures, occlusions and wide rotations. [Wang 07] show results for rotations below 30°. Most of the literature using this approach is focused on applications requiring low resolution images, such as detecting meeting room interlocutors interaction [Cordea 01, Xiong 05, Canton-F. 07].

Tracking methods, whether only tracking or hybrid systems, provide better accuracy than previous approaches. This technique is user independent, and its implementation can easily meet real-time requirements. Examples are [Xiao 02, Oka 05, Zhao 07] among others, which have errors below 3°. [Xiao 02] presents results only for rotations up to 75°, with an accuracy of yaw about 3.8°. In a recent publication [Sheerman-C. 09] presented an online learning model proposal, achieving 3.8° and 4.2° error for pitch and yaw rotations. However, their results were only evaluated in a range of ±40° and ±20° respectively. The same way, [Oka 05], while showing very good results, with an error as low as 2° for yaw rotations, only evaluates the systems for sort sequences, and small rotations. They create an static model at the initialisation step, so no wider rotations are possible. It is not clear how well the system can deal with the drifting problem, for longer video sequences. Many use SIFT or SIFT-like features [Yang 02b, Ohayon 06, Zhao 07], however, the low light conditions in a simulator are not appropriate for SIFT-like matching techniques, as it will be shown in the results on chapter 7.

The more prominent results are obtained using a 3D face model [Wu 00, La Cascia 00, Xiao 02, Krinidis 09]. Having the possibility to use a stereo rig, this is probably the solution that provides the best accuracy. Using a 3D face model notably improves robustness, since it makes possible to detect tracking errors due to similarity appearance of different parts of the face under some rotation. Some authors have used generic dense face models, such as cylindrical [Lin 10] or ellipsoidal [An 08] ones, and the whole face or feature textures are mapped to the model shape. However the wider rotation ranges are provided by sparse models formed from feature 3D coordinates.

Despite the variety of related works, the face tracking problem is still open, and none of the detailed solutions deal with the problem of having at the same time a full-range, accurate, user independent, real-time and calibration free pose estimation system. Many of the model-based systems rely on generic models, which do not fully adapt to individual geometry. On the other hand, other methods, based on appearance and requiring train-

ing, do not generalise well enough to be classified as user independent. A dynamic 3D model can be fitted to any user and give an accurate estimation while being user independent. However, it needs being updated under different user poses, both in geometry and appearance, in order to maintain performance on the full rotation range and illumination changes.

2.4 Aim of the thesis

The aim of this thesis is to develop an algorithm to accurately estimate on real-time the driver's gaze, focusing on the pose of the driver's face. This system will be used in practise to analyse, from the gaze fixation information, driver's distractions inferred in a naturalistic simulator. The results generated for pose and gaze estimation and derived data such as distraction statistics will be used in the simulator as a tool to study distractions and driver's behaviour inside the cabin. The developed method must be able to detect and track the driver's face over wide rotation angles, under low light conditions and accurate enough to allow for a further study of the driver's gaze, creating an automatic and adaptive 3D face model online.

After the review of the State of the art and considering the requirements presented in the introduction, the aims of this thesis are as follows:

1. To study the tracking techniques for low light conditions and wide tracking range.
2. Using an automatic 3D model adapted to each user, and online refinement during execution.
3. To develop an algorithm capable of an accurate calculation of the face pose and gaze.
4. To generate coarse fixation information on the environment —scene and cabin— to infer driver's distraction behaviour and statistics.
5. To evaluate the proposed algorithm performance using a video dataset and its ground-truth.
6. To design different exercises focused on driver's distraction in a naturalistic simulator and study the adequateness of the proposed system for a group of psychologists to extrapolate information from the designed exercises.

Chapter 3

Face Pose Estimation Architecture

Following the discussion of the State of the Art, this chapter presents the general architecture of the proposed face pose estimation algorithm.

The face pose estimation approach in this thesis is based on tracking methods, since it was shown that they obtain the best accuracy. The system here presented is based on tracking a set of features which are automatically detected on the subject's face, with a calibrated stereo rig. Since one of the requirements of the system is that it must be user independent, the feature selection process does not use *a priori* information. Instead, the features are selected upon high contrasted regions of the face with an interest point detector.

Many of the tracking approaches reviewed above use a face or head 3D model. Some apply a model which coarsely approaches the form of the head, such as a cylindrical model, while others build a more precise 3D model adjusted to the face surface. Here, the features are arranged in the form of a sparse 3D face model, using the 3D coordinates of each feature, obtained from a stereo rig. The model allows for a better feature tracking support: it makes possible to detect errors in the location of the features in the images, if any of them are far from the expected projection of the 3D model points. These detection errors can appear by the similarity of the features to different parts of the face under rotation or illumination changes. This kind of errors are much harder to detect using a 2D model. Our proposal is also able to adapt the model over time, and it works over the full range of head rotation and under low-lighting conditions.

3.1 General architecture

The algorithm presented in this thesis is designed to automatically extract the interest points and to build the 3D model of the face, just requiring the driver to look straight ahead at the initialisation frame.

To follow features appearance variations due to rotation, a feature template selective re-registering technique is carried out using a novel mixed-views technique using both cameras. This way, one camera is used to anticipate what the other will see, whereas this other camera is used for tracking, avoiding appearance variations of the features due to the projection on the 2D image plane. During yaw rotations, the selective re-registering chooses the frames in which pose uncertainty is minimal to avoid the template drifting problem. For roll and pitch rotations, a feature warping is performed to diminish the projection variation. Incorrectly tracked points (outliers) are detected based on their Euclidean distance to the model point projections after pose estimation, and discarded using

a RANSAC [Fischler 81] process. In addition, a pose uncertainty can also be estimated based on the sum of this Euclidean distance for the inliers. 3D pose is recovered from the set of 2D points assuming weak camera projection, using POSIT [Dementhon 95] or the Levenberg-Marquardt algorithm [Lourakis 04]. Finally a bundle adjustment algorithm (BA) [Triggs 99] is used to correct the model.

Initially the model only contains features from a frontal face, which will self-occlude under wide rotations. To increase the range of rotation, it is extended with the addition of new features, when the number of non occluded ones falls bellow a minimum. The same way the initial model creation is completely automatic, the model extension is also automatically performed when the algorithm requires it, and the conditions are appropriate for this. However, the 3D coordinates of a new added feature heretic the specific error related to the poses in which the feature is being added. To correct this, a bundle adjustment background process constantly corrects the model 3D points at some key-frames. This allows for accurate point addition to the model, and the algorithm works reliably for the whole yaw rotation range, $\pm 90°$ degrees.

The main blocks of the system architecture are shown on figure 3.1 and can be summarised as follows:

(a) Initially, a sparse 3D model is automatically built with features extracted from subject's face using a stereo rig.

(b) From frame to frame, the model pose is estimated from the features located applying a novel camera mixed-view re-registering approach.

(c) At some key-frames, re-registering is performed. The 3D model might be extended to previously occluded parts of the face and corrected with bundle adjustment.

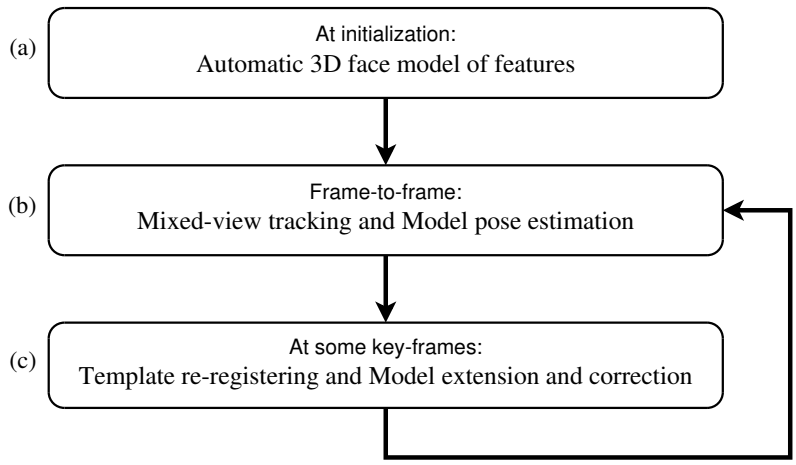


Figure 3.1: Main blocks of the face pose estimation algorithm.

On model creation, the initial set of features is chosen based on their saliency, so their frame to frame matching fails with low probability, minimising localisation errors in subsequent steps. In this sense, three techniques are tested as interest point detectors: SURF [Bay 06], Harris [Harris 88], and multiscale Harris [Triggs 04]. For this purpose, the face is initially detected on the frames of both cameras using Viola & Jones algorithm.

Once the features are selected and their 3D coordinates recovered using epipolar geometry restrictions, they must be filtered to reject those which have clearly been incorrectly

stereo matched. A geometric constraint of maximum face size is also applied. In addition, some authors demonstrated that the face can be assimilated to a cylindrical model [La Cascia 00, Xiao 02, Lin 09]. Here, a horizontal-oriented cylinder is fitted to the face using an optimisation method, and those features which are far from the cylinder surface are also rejected. This way, the model can include points from salient parts of the face (nose, ears, etc.), which are not on the cylindrical surface that includes the majority of points. This relaxed contraint allows to create a more realistic model, and represents an improvement with respect to previous works in the State of the art. Furthermore, we use the information obtained from the fitted cylinder to estimate the angle of each feature to the cylinder geometric centre, information which can be used to estimate self-occlusion.

Self-occlusion is a drawback of creating the 3D model from a initial single pair of frames. Taking into account that the required rotation range for the system is $\pm 90°$ in yaw angle, under a wide rotation some of the features detected with a rotation of $0°$ will surely be hidden behind part of the face or head, making the system lose track of them. The accuracy of the pose estimation depends on the number of features, and thus a model extension procedure is needed. New features are added to the model when all of the next conditions are met:

1. New parts of the face are exposed to the cameras.
2. Pose estimation uncertainty at the current frame is low.
3. Bundle adjustment has finished correcting the 3D coordinates of previously added points.
4. The number of visible features is higher than a minimum, to ensure algorithm robustness.

Since no other *a priori* information is used at the initialisation step, the face rotation at initialisation represents the pose rotation reference, and it is arbitrarily assigned a rotation of $0°$, with unitary director vector $\vec{u} = (0, 0, 0)$. Following rotation estimations are relative to this reference. If at this initial frame the actual user rotation is different from zero, the reference includes an offset error, and all the subsequent estimations show the same offset.

All these steps are executed in a completely automatic way. The only previous process required is the initial stereo calibration of the stereo camera rig. An accurate calibration improves the 3D points estimation, model accuracy, and consequently error correction. Figure 3.2 shows the general architecture of the proposed algorithm. It consists of the following steps:

0. **Initial system layout (offline):** A calibrated stereo-rig is placed in front of the user, looking at the subject's face, typically between the driving wheel and the wind-screen of the vehicle. This step is only carried out once when the cameras are installed. All other operations are fully automatic.
1. **Face detection:** User's face is detected on the initial image of both cameras using the Viola & Jones algorithm.
2. **Interest points detection and features extraction:** An interest point detector algorithm is applied on one of the cameras in the area where the face has been detected.The interest points are matched over the two cameras' images using epipolar geometry restrictions to establish putative pairs used to calculate the 3D coordinates

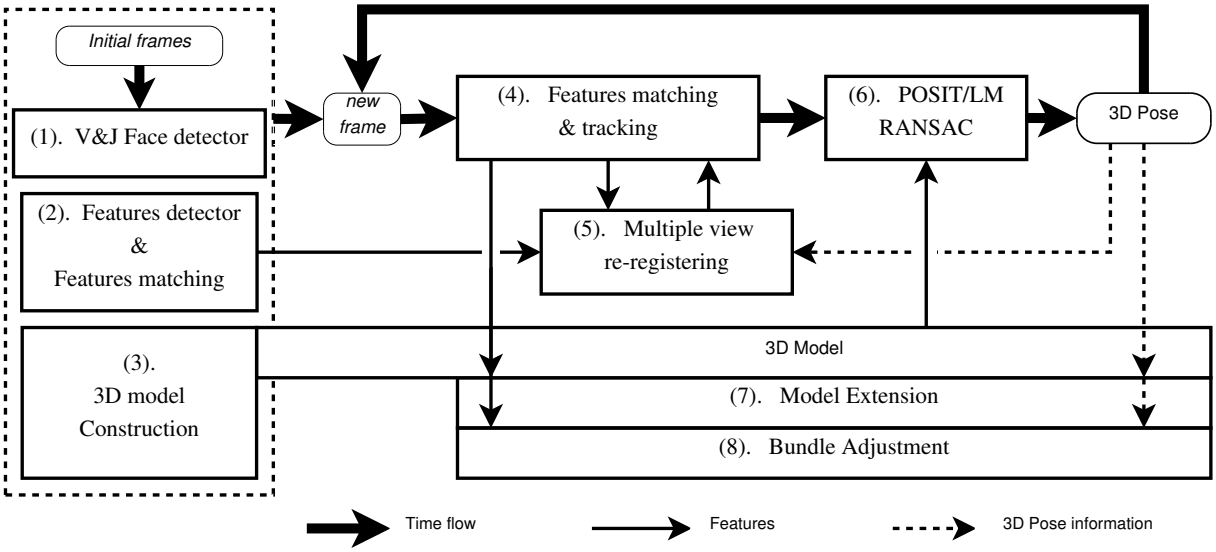


Figure 3.2: General architecture of the face pose estimation algorithm.

of the detected features. The interest points that do not match or do not meet the required quality are discarded.

3. **Automatic model construction:** An initial 3D face model is created automatically. The model is defined by a set of features, each composed of the 3D coordinates of the feature point and a patch or descriptor containing the feature texture. The model's origin, to which subsequent pose estimation is relative to, is set to the 3D geometric centre of all the features, and all of them are moved rigidly with the model. The initial rotation reference is set to 0° and initial translation reference to a zero vector. Model points are filtered attending to some geometrical restrictions.

4. **Features matching and tracking :** Model feature points are tracked frame to frame on the image of the same camera. No stereo information is used for tracking. Correspondences are obtained via thresholded matching between stored patches of the model containing the feature templates and the patches from the current image.

5. **Features re-registering:** On demand, when the proper conditions are met, the stored feature patches of the model are updated to follow appearance variations. A clustering of multiple-view patches acquired from both cameras selects the best one to be used for matching depending upon rotation.

6. **Pose estimation:** The pose of the face is estimated from the feature locations with either the POSIT algorithm or Levenberg-Marquardt, within a RANSAC process to reject unsuccessfully tracked points or outliers derived from the tracking step. The 3D face model is used by RANSAC to guide the correct point distribution.

7. **3D Model extension:** When rotation takes place, if conditions are met, the 3D model may be extended with new features detected over the face using the same technique described at step 2, to include previously occluded parts of the face in the model.

8. **Bundle adjustment:** A background bundle adjustment optimisation process refines the model, including the existing and the newly added 3D points of the model.

This process is executed in parallel to the rest of the algorithm, at some key-frames, and stopped when no further corrections are required.

Chapter 4 presents the model creation algorithm. Feature tracking, pose estimation and model correction are described in chapter 5.

Chapter 4

Automatic 3D Face Model Creation

The face pose estimation system presented in this thesis is based on the tracking of a set of face features structured as a sparse 3D model. Although the 3D model formation takes place during the whole execution of the algorithm, there is an initial *draft* model which is automatically created on the first frames of the algorithm. After initialisation, the model will continuously be improved with corrections during execution, and extended with new features. The purpose of this model is to track the user's face in a robust way and to provide a reference from which the pose can be extracted from 2D feature projections on the camera images.

Figure 4.1 depicts the different steps involved in the model creation. The model comprises of the 3D coordinates of features and a cluster of its appearance descriptors associated with each feature, which are used for 2D tracking and later pose estimation.

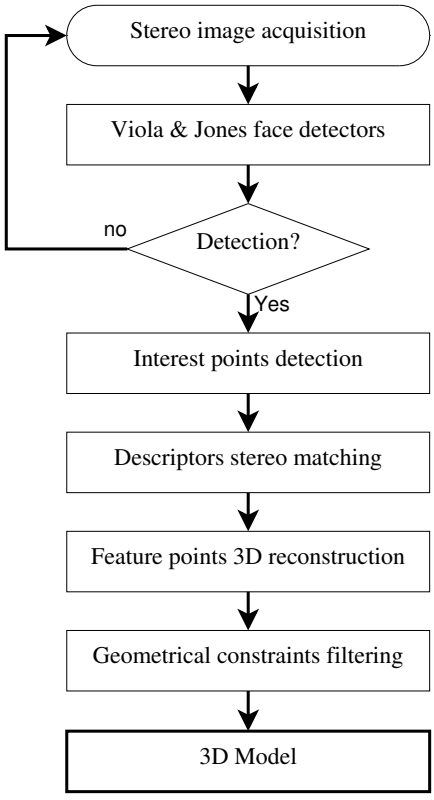


Figure 4.1: General layout of the model creation process

There is not any prerequisite about the election of the features. They will be interest points from the point of view of the detection algorithm.

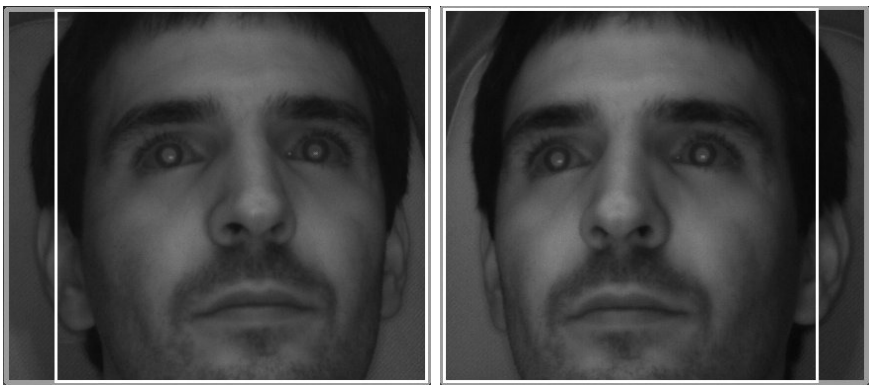
The rest of this chapter shows some comparisons between different methods to obtain the model features and discusses the technique used in each step of the model creation.

4.1 Initial Features detection and stereo matching

To create the model, features must be detected within the bounds of the face. The first step of the process is to detect a frontal face, using the Viola & Jones algorithm in the right and left initial camera frames.

At this step, the user is asked to look forward, right to the centre of the stereo rig, so the V&J can detect an almost frontal face in both images. This will be the only initialisation process the user will be asked to perform. V&J loops frame to frame until the face is detected in both images. The current frames are set as the initialisation images, I_0^r and I_0^l.

Typically, V&J detects a bounding box that can leave outside its bounds part of the face, e.g., ears, specially when the face exhibits a small yaw angle with respect to any of the cameras, as we can see in figure 4.2. Due to the base line of the stereo cameras, this is sure to happen at least for one of the cameras, if not for both. This effect can lead to a small part of the face located outside the initial area for feature extraction, and consequently not generating interesting 3D points for the model in the rejected area. Although the model will be increased later, as it will be explained in Section 5.4, it is desirable that the initial model creation encloses as much part of the face as possible. For this purpose, the detected V&J bounding box is widened 50 pixels to the left on the right camera image, and to the right for the left camera image, to ensure that the whole face is within in both cases. This value has been obtained experimentally. On the contrary, it is not widened up-down since from experimental results it have been found that the jaw and hair do not provide much reliable information to the model. This widened box is the detection area of the face where the features are searched for. Figure 4.2 depicts the original V&J and widened detection box.



(a) I_0^l (b) I_0^r

Figure 4.2: Viola & Jones detection box (inner box) and 50 pixels widened feature search area (outer box). The widened search area includes part of the face that V&J leave outside its detection box

4.1.1 Features extraction and matching methods

Once the face has been located in the images, the next step is the feature extraction process. A feature i is represented by its appearance template or descriptor, $\mathbf{T}_i$, its 2D position on both camera images, $\mathbf{x}_i^{\{r,l\}}$, and its 3D coordinates, $\mathbf{X}_i$. The 2D position in the images actually can be computed as the projections of $\mathbf{X}_i$ over each camera as follows:

$$\mathbf{x}_i^{\{r,l\}} = H^{\{r,l\}}\mathbf{X}_i, \tag{4.1}$$

where H^r is the projection matrix to the right camera image, and H^l to the left one. The template or descriptor is extracted from a patch on one or both camera images, located around $\mathbf{x}_i$.

To obtain a feature i, it is first necessary to obtain its 2D projections on each camera, establish the correspondence of $\mathbf{x}_i^r$ to $\mathbf{x}_i^l$ and then compute $\mathbf{X}_i$ by stereo recovery. Each $\mathbf{x}_i^{\{r,l\}}$ is obtained from a set of *interest points* in the image. These interest points represent parts of the image which are likely to be easily matched to their counterpart interest point on the other camera image, and on subsequent frames over time. Consequently, interest points must be easily differentiable from other parts of the image. These differentiation process involves the point extraction itself, and the process of establishing the correspondences to the same point on other image, that is, the *matching* process.

The matching process relates two interest points in two different images. If for any reason, an interest point can not be matched with any other from the other image, it is discarded and consequently does not become a feature. Although the extraction and the matching are two different subsequent processes, they are strongly related, and are considered and studied together in the literature. Each matching technique comes along with its own extraction algorithm.

To generate the features which will form the model, interest points are extracted over the widened V&J boxes on the right or on both images, depending on the technique used. As a system requirement, described in section 1.5, the feature extraction process must be user independent, i.e., it must not use any a-priory information.

Different authors have published comparatives on detectors [Mozos 07, Moreels 07] and image registration methods [Zitova 03, Brown 92]. Most of them cover the general case of well defined objects, full of corners and normal lighting conditions. The following sections study the appropriateness of different detectors and matching techniques existing in the literature to find the stereo correspondences of the face features under low light conditions.

SURF features and Matching

Speeded Up Robust Features (SURF) is a fast scale-invariant feature detector and descriptor. It was introduced by Bay *et al.* in [Bay 06] and revisited in [Bay 08]. SURF features can be used in computer vision tasks like object recognition or 3D reconstruction. Inspired by the highly influential *Scale-Invariant Feature Transform* (SIFT) [Lowe 99], the standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations. SURF is based on sums of approximated 2D Haar wavelet responses and makes an efficient use of integral images (see [Viola 04]). The main advantage of SURF over SIFT and other competitors, relies on the fact that SURF features can be computed faster due to the use of integral images, allowing a faster matching.

In order to apply SURF, first many interest points are extracted from the region of interest, the face search box, from both camera's images. Since an interest point must be detected on both images to establish a correspondence between them, it is important for the detector to offer a high *repeatability rate*. The *repeatability* measures the reliability of a detector for finding the same physical interest points under different viewing conditions.

SURF uses a *Fast-Hessian detector* to locate interest points. They are detected at locations where the determinant of the Hessian matrix is maximum. Given a point $\mathbf{x} = (u, v)$ on an image I, the Hessian matrix $\mathcal{H}(\mathbf{x}, \sigma)$ in $\mathbf{x}$ at scale σ is defined as

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{uu}(\mathbf{x}, \sigma) & L_{uv}(\mathbf{x}, \sigma) \\ L_{uv}(\mathbf{x}, \sigma) & L_{vv}(\mathbf{x}, \sigma) \end{bmatrix}, \tag{4.2}$$

where $L_{uu}(\mathbf{x}, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial u^2} g(\sigma)$ with the image I at point $\mathbf{x}$. The term $\frac{\partial^2}{\partial u^2} g(\sigma)$ can be approximated at a very low computational cost using integral images [Simard 99]. Integral images allow for fast computation of box type convolution filters. The entry of an integral image $I_{\sum}(\mathbf{x})$ at a location $\mathbf{x} = (u, v)$ represents the sum of all pixels in the input image I within a rectangular region formed by the origin and $\mathbf{x}$.

$$I_{\sum}(\mathbf{x}) = \sum_{i=0}^{i \leq u} \sum_{j=0}^{j \leq v} I(i, j). \tag{4.3}$$

Gaussians are optimal for scale-space analysis [Lindeberg 98], but in practise they have to be discretised and cropped. This leads to a loss of repeatability under some image rotations. However, since in this case the two images are taken from the calibrated stereo rig at the same time, they are known not to have any rotation one to the other.

Interest points h_i must be found on $I_t^{\{r,l\}}$ at different scales in order to preserve repeatability, and are extracted after a non-maximum suppression in a $3 \times 3 \times 3$ in the space-scale neighbourhood [Neubeck 06] after applying the Fast-Hessian detector. The 64-dimensional descriptors are calculated within an interest points neighbourhood from the first order Haar wavelet responses in u and v directions. The area depends on the scale being used. Fast matching of interest points is performed by means of Euclidean distance between two 64 dimension descriptors. For each interest point in an image, only those which are proximal to its epipolar line on the other image are compared. An interest point h_i^r in I_0^r is matched to other h_j^l in I_0^l if it satisfies

$$j = \arg\min_k (\mathbf{T}_i^{(r)} - \mathbf{T}_k^{(l)}), \tag{4.4}$$

where $\mathbf{T}_i^{(r)}$ and $\mathbf{T}_j^{(l)}$ are the SURF descriptors for interest points i and j on I_0^r and I_0^l respectively. For improved robustness, the matching process is performed as well in the opposite direction, and if i and j are not the same, then the interest point is rejected. That is, in minimisation of equation (4.4), now let k be the sub-index of the term $\mathbf{T}^{(r)}$:

$$i_2 = \arg\min_k (\mathbf{T}_k^{(r)} - \mathbf{T}_j^{(l)}) \tag{4.5}$$

The resulting i_2 must be the same i than in equation (4.4). Figure 4.3 depicts all the interest points detected on a face, while figure 4.4 shows the stereo correspondences for the face features obtained from the successfully matched interest points.

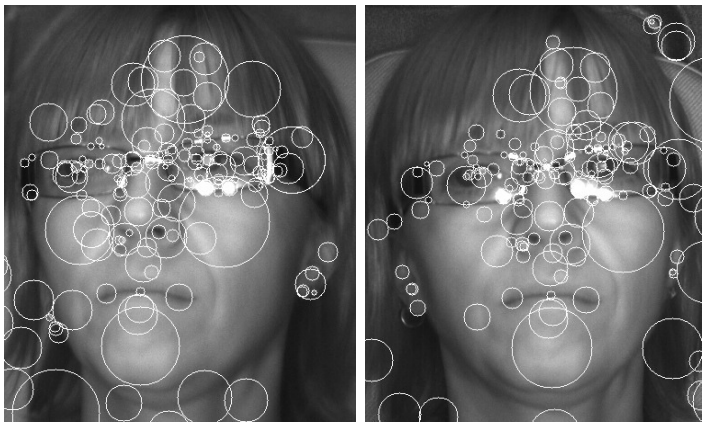


Figure 4.3: SURF detected interest points for right and left camera images

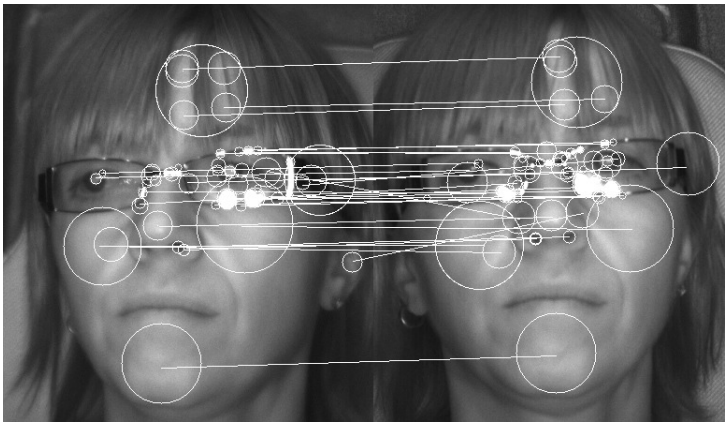


Figure 4.4: Stereo correspondences of face features obtained using SURF

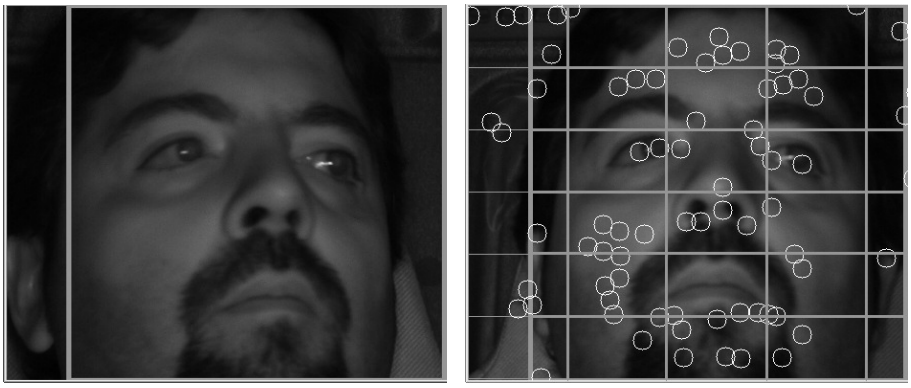
Harris detector and cross-correlation matching

The Harris corner detector [Harris 88] is a popular interest point detector due to its strong invariance to rotation, scale, illumination variation and image noise [Moreels 07]. Without going into details, it is based on measuring the variations of the autocorrelation function of the intensity image, $c(u, v)$, under small patch shifting in vertical and horizontal directions. Interest points are extracted at image points where the variation of the autocorrelation function has local maximums. This indicates that $c(u, v)$ changes significantly for both Δu and Δv, meaning that the position (u, v) is a corner. For a wider explanation and Harris detector equations, we refer the reader to [Harris 88] and [Moreels 07].

In order to achieve a spread distribution of initial features candidates, the face on image I_0^l is divided into a regular grid as it is shown in figure 4.5(b), and each cell is required to contain two to five interest points. If a frame does not fulfil this condition, the stereo frame set is rejected and the algorithm loops, starting all over again with the V&J over a new frame set, until conditions for model creation are met.

Let $h_i^r = (u_i^r, v_i^r) \in \mathbb{R}^2, i = 1 \dots n'$ be the n' candidate points on I_0^r, derived from Harris corner detector, as shown in figure 4.5(b). Correspondences h_i^l on I_0^l are obtained via normalised cross correlation matching of the 2D patch around h_i^r over its corresponding epipolar line on I_0^l.

A point is considered matched to its correspondence if the correlation maximum is higher than a threshold, initially set to 0.5. Only N_0 points (out of the set of n') will



(a) I_0^l and Viola & Jones box (b) I_0^r and Harris points, $h_i^r,\ i = 1 \dots n'$

Figure 4.5: Viola & Jones boxes for left (a) and right (b) camera images, and initial distribution of feature candidates $\{h_i^r\}$ detected using Harris (b).

be valid since some points will be discarded due to incorrect matching and/or wrong 3D point estimation.

Multiscale detector and matching

Although the Harris corner detector is reasonably scale invariant, sometimes there are interest points in an image which are detectable in a certain image scale, but not in others. To maximise the number and quality of interest points detected, Harris can be applied to detect scale-space features [Baumberg 00]. This technique extracts a set of interest points at different image scales, and associates the scale with the point. The matching step is then performed at the corresponding scale at which the point was detected.

The power of multiscale image analysis comes from the ability to choose the resolution at different parts of the image dynamically. The image area around the eye, for example, presents more details than cheeks. Thus, using different scales it is possible to extract proper features from both sharp and smooth parts of the face. This is very similar to the idea below the SURF feature detector explained above. But in this case, the feature extraction is done by means of a Harris Detector, using the same face image at different resolution scales.

To apply a multiscale Harris, a Gaussian pyramid reduction process is computed to resize the image. This process involves lowpass filtering and downsampling the image pixels. At each step of reduction, first the source image is convolved with a Gaussian filter and then it is downsampled by a factor of two rejecting even rows and columns. The reduction of an image I to a smaller image f_k using a Gaussian kernel of size s can be expressed as

$$f_0 = I_0, \qquad G_k = f_k g_s(m,n) = f_k e^{-x^T x/2\sigma^2},$$

$$f_k(u,v) = \sum_{m=1}^{s}\sum_{n=1}^{s} g_s(m,n) \cdot G_{k-1}(2u+m, 2v+n) \tag{4.6}$$

where $g_s(m,n)$ is a squared Gaussian kernel of size s. Figure 4.6 shows a scheme of this process, and figure 4.7 depicts the results applied to driver's face.

Generally the same characteristic is detected at different scales, and so all the local maximums at different scales must be related to a single interest point. [Baumberg 00] solved this by ordering the maximums in a neighbourhood in function of its strength at

different scales, and comparing the vector of maximums on the image being correlated. Since for the specific problem of establishing stereo correspondences only one scale per feature is needed, as there is no scale variation from left to right image, in our case we apply a neighbourhood maxima suppression on the scale-space dimension to obtain the interest point at the strongest scale.

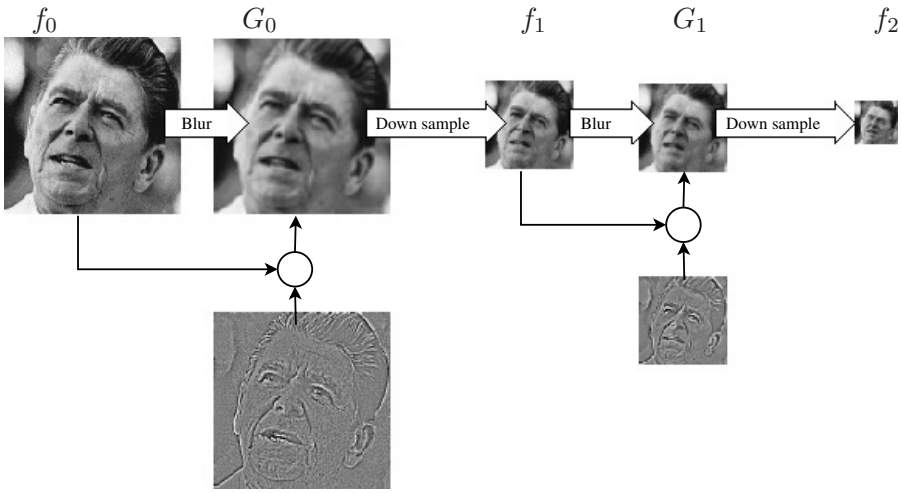
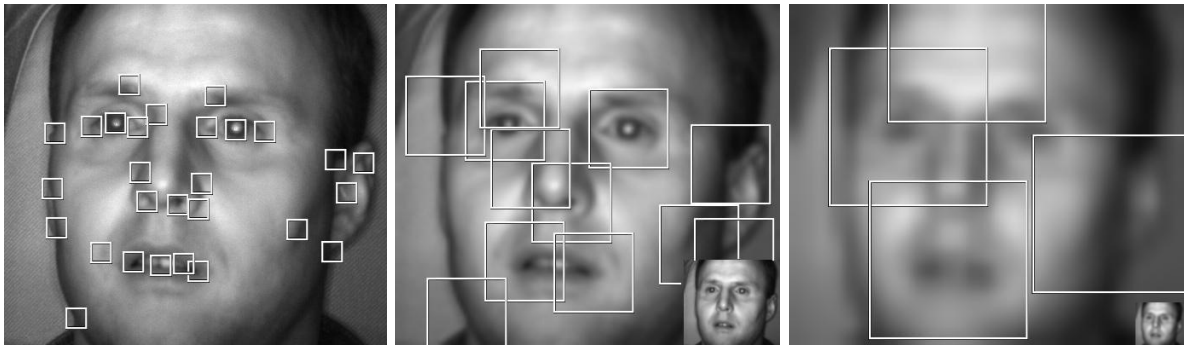


Figure 4.6: Decomposition step for two-level Gaussian Pyramid. The finished pyramid for each scale step consists of the images $\{f_0, f_1, f_2\}$.



(a) Original image

(b) Image scaled down twice (total scaled down by a factor of 4)

(c) Image scaled down three times (by a factor of 8)

Figure 4.7: Multiscale images of a driver's face and detected features at each scale

Results and Discussion

Stereo correspondences of a feature are obtained by means of a stereo matching process, using any of the three methods previously described. Features are extracted from the face in one or both camera images and their correspondence searched. Each of the stereo matching techniques is associated with a feature detection technique, and they are not generally interchangeable.

The overall performance of the different methods can be tested in terms of the number of correct correspondences. For each interest point, there are three possibilities. In the first case, it is not matched to any other interest point or descriptor in the other image because of the distance in the descriptor space (SURF descriptor or appearance) is higher than a

required threshold. The interest point is rejected. In the second case, based on descriptor distance, it is matched to other feature, but the matching is incorrect, i.e., the match does not correspond to the same feature on both images. This is called a *false positive* or *false alarm*. In the third alternative, a match is established and the correspondence is correct. This is called a *detection*. The *detection rate* is the ratio between detections and the total number of extracted interest points.

Using data extracted from the ground-truth (obtained as explained in section 7.2), we have measured the repeatability rate of SURF and the general performance of the studied matching techniques under the conditions present in the simulator. Figure 4.8 shows the detection rate as the number of extracted interest points increases. To collect these data several users have been tested under different illumination conditions, and the number of incorrect correspondences have been hand-marked. Inside the simulator cabin, part of the illumination is produced by the light coming from the projection panels which simulate the road and environment. This light is back-projected to the panels located in front of the cabin, and slightly illuminates the user's face. To increase this illumination, IR leds are arranged around the cameras. However, it is known that long exposure to IR illumination causes fatigue to the driver's eyes [Agilent 99, Koons 03], and consequently the IR kept as low as possible.

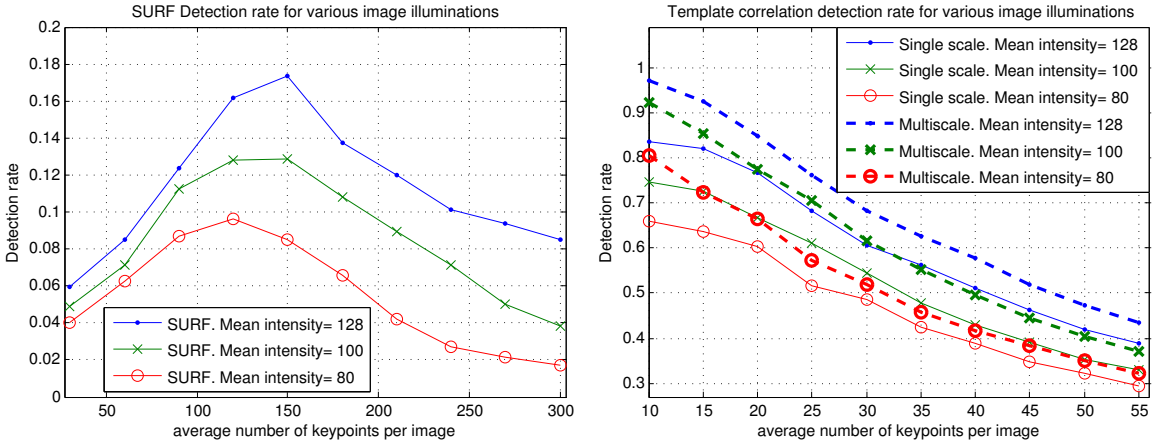
For SURF detection, the number of interest points is changed by adjusting the thresholds for the Hessian image in equation (4.2). Graph 4.8(a) shows that best performance is obtained when around a hundred and fifty interest points are extracted. Note that the ratio rapidly decreases when the mean intensity illumination in the images gets poor. This means that SURF is not an appropriate technique to be used for images taken in a driving simulator. Graph 4.9 shows the detection rate using Harris plus template correlation techniques, at a single scale or multiscale. It can be observed how multiscale performs slightly better than single scale. In this case, the number of interest points is increased by reducing the minimum quality required for detection in the Harris algorithm.

ROC curves, in figures 4.9(a) and 4.9(b) also show that Multiscale matching slightly outperforms the other techniques in terms of number of false alarms, although the different is not remarkable.

Typically, face features are not as highly contrasted or textured as those used in other 3D structure-from-motion applications, like for example [Davison 07]. A face does not typically present corners, and features are exposed to illumination variations and shines. Moreover, the limited illumination leads to dark images, where features are even less contrasted. Another important issue derived from low illumination conditions is the poorly focused images. The iris of the camera must be wide open to allow the maximum light into the sensor, which drastically reduces the depth of field. This produces that some parts of the face, specially if the driver moves forward or backwards, might not be well focused.

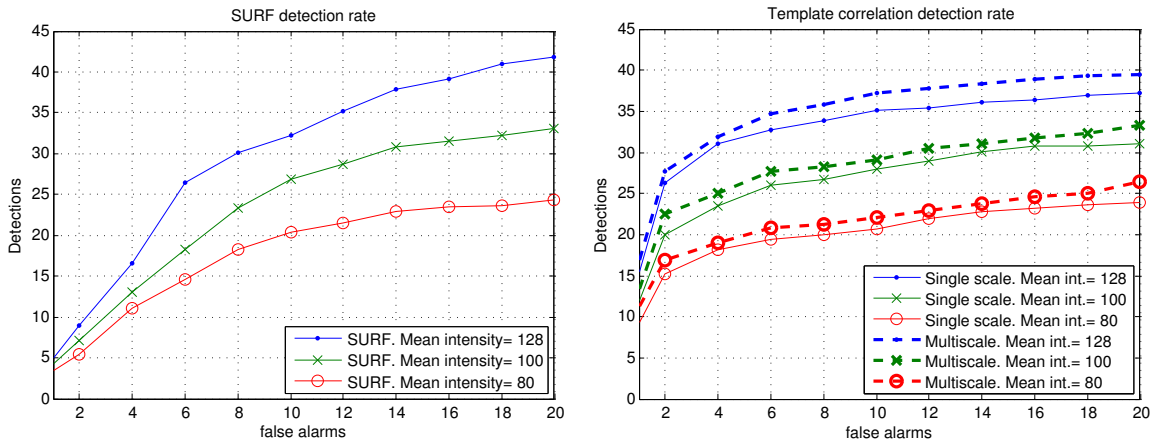
All this makes that a technique such as SURF descriptors does not give good results, mainly because the feature candidates extraction fails to choose correct candidates for matching due to the absence of corners. Incrementing the number of feature candidates for matching directly affects computational cost and real time performance.

SURF uses a fast interest points detector and a fast matching technique to compare many candidates found on both images. The repeatability rate is an important parameter, since a correspondence over a feature can only be established if the same feature is detected in both images. If this happens, the matching is more likely to be positive for wider rotations than using template correlation, since the SURF rotation invariance is better. In the case of correlation, the repeatability is not an issue, since in this case, interest



(a) SURF (b) Template correlation

Figure 4.8: Comparison of the detection rate using SURF or template correlation, versus the total number of extracted interest points on both images. Results are shown for various image mean intensity



(a) SURF (b) Template correlation

Figure 4.9: Comparison of the detection rate using SURF or template correlation, versus the false alarm rate. Results are shown for various image mean intensity

points are not extracted in the second image. Instead, these are only obtained in one of the images, and the features searched within a predefined area on the other. On the other hand, template correlation is slower than comparing two SURF descriptors, so less candidates are used for matching.

Although the model will be extended during execution, the correctness of this first step of model formation is vital for a later correct and accurate face pose estimation process. The 3D model should initially be composed with at least 20 features, and preferably 25, to have a consistent coarse model for face pose estimation. If the images have mean intensity level of 80, in a scale from 0 to 255, the graphs on figure 4.9 show that using SURF, the false alarms reach an average of 10 to obtain at least 20 correct features, and it is difficult to get 25 correct ones. Using Multiscale correlation the situation is not much better. It is only possible to get enough correct features at the cost of a high false alarm rate.

Thus, the technique used for matching in this thesis is a modified template correlation,

based on using various patch sizes, instead of different scales.

4.1.2 Multisize matching proposal

Feature matching performance is sensible to different effects depending on the size of the patches used. If a feature changes its appearance because of projection, it works better to match small patches of the image, for which the changes will be more homogeneous than for bigger ones. On the other hand, if the image is not well focused, using bigger patches is more adequate, in order to reduce the number of incorrect matches due to repetitive texture patterns in the face.

As a convenient solution, we characterise each feature texture by three different size patches centred on the feature. The matching method is based on the addition of the matching result for the three different size visual patches.

Let $h_i^r = (u_i^r, v_i^r)$ be a feature candidate or interest point on the initial right image, I_0^r, and $P_{ik}^r = P(h_i^r, \mathbf{s}_k)$ be a patch on I_0^r around h_i^r of size $\mathbf{s}_k \in \mathbb{R}^2$. To find its correspondence h_i^l on image I_0^l, three patches of different sizes are defined, P_{ik}^r, $k = 1 \ldots 3$. Then, the three patches are matched over a search area of size $\mathbf{s}_{search}$ on I_0^l, producing matching results $r_{i,k}^l(u, v)$ respectively, all of them of the same size. The search area $\mathbf{s}_{search}$, and consequently the size of the correlation results, is defined as a region seven pixels wide around the epipolar line on I_0^l corresponding to the point h_i^r, and it is independent of the size $\mathbf{s}_k$ of the patch. The correlation result is expressed as

$$r_i^l(u, v) = \sum_{k=1}^{k=3} r_{i,k}^l(u, v). \tag{4.7}$$

The template matching problem can then be formulated as finding the location (u, v) in the image I_0^l that maximises the objective function,

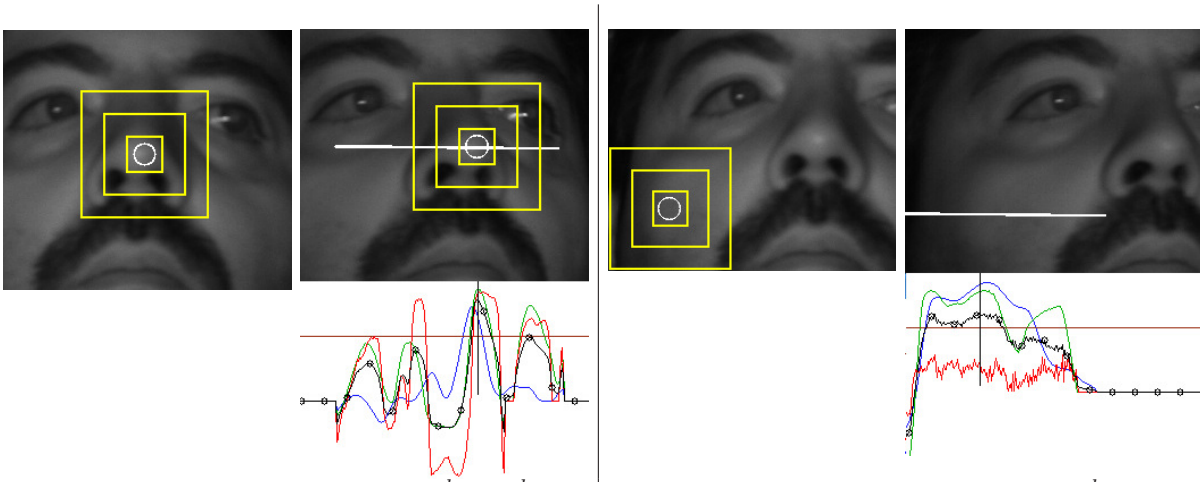
$$h_i^l = (u_i, v_i) = \arg\max_{u,v}(r_i^l(u, v)). \tag{4.8}$$

To ensure the robustness of the feature and to minimise matching error, candidate points which do not meet the condition

$$r_{i,k}^l(u, v) > h_{tc}, \qquad k = 1 \ldots 3, \tag{4.9}$$

are rejected, where h_{tc} is the matching threshold, set to 0.5 at the stereo matching step. This restriction helps reducing the number of false alarms.

Figure 4.10 depicts the matching results for two different interest points. (a) is correctly detected and matches its correspondence, while (b) depicts a failed matching and consequently the interest point is rejected. The three concentric boxes in (a.1) and (b.1) are the three patch sizes for the texture on I_0^l. These are the patches which are correlated over the search area in (a.2) and (b.2) to find the stereo correspondence location on I_0^l. Since we are looking for stereo correspondence, the search area is restricted to the epipolar line (horizontal line in the images) of the interest point. The three graphs in (a.2) represent the correlation result of the three patches over the epipolar line, being the x axis the u coordinate in the image, and y the intensity of the correlation. The curves on figure (a.2) shows how local maximums lay approximately at the same u pixel location. The matching result is given by the maximum of the black circle marked line. The better patch size to be chosen will be discussed in section 7.4. Here, the sizes chosen are $\mathbf{s}_1 = 25 \times 25$, $\mathbf{s}_2 = 61 \times 61$ and $\mathbf{s}_3 = 91 \times 91$.



(a.1) h_i^r on I_0^r (a.2) h_i^l on I_0^l (b.1) h_j^r on I_0^r (b.2) I_0^l

Figure 4.10: A feature candidate h_i^r on image I_0^r (a.1), and its correspondence h_i^l on image I_0^l over the epipolar line (a.2). The graph in (a.2) shows the results of $r_{i,k}^l(u,v)$, $k = 1 \dots 3$ and $\mathbf{s}_1 = 25 \times 25$ (red), $\mathbf{s}_2 = 61 \times 61$ (green), $\mathbf{s}_3 = 91 \times 91$ (blue), and $r_i^l(u,v)$ (black), all restricted the epipolar line. (b) Matching is not correct and point is discarded.

Multiple instances of patch matching, one for each camera, are run simultaneously, frame by frame. These two processes do not currently interact, but it might be possible to improve the accuracy by implementing a mixed camera matching in the future.

Results and discussion

By using the patch correlation at three different sizes, the feature matching process is optimised for a structure-from-motion application where we wish to ignore unreliable matches at the expense of reducing the number of feature matches. For this purpose the newly designed multisize matching method applies much restrictive conditions to validate a correspondence.

Although this proposal has the side effect of reducing the total number of extracted features, these are more reliable, allowing for a better subsequent outlier detection and filtering of the incorrect ones. Figure 4.11 shows the detection rate using the Multisize template correlation. In this case, for an image average intensity level of 80, to obtain 25 features in the model, the false alarm rate is around 8, lower than using the previous methods.

4.2 3D Face Model

Given the stereo correspondences calculated in the previous step, the next one is to calculate the 3D coordinates of the features.

Using stereo equations and calibration parameters of the stereo rig, it is possible to calculate the 3D coordinates of a feature, knowing its 2D projection points on the two camera images. This process is known as the stereo 3D reconstruction, although the algorithms to solve this are also called *triangulation algorithms*. In this thesis, we use an invariant method to the projective space. The equation between a 3D point and its 2D

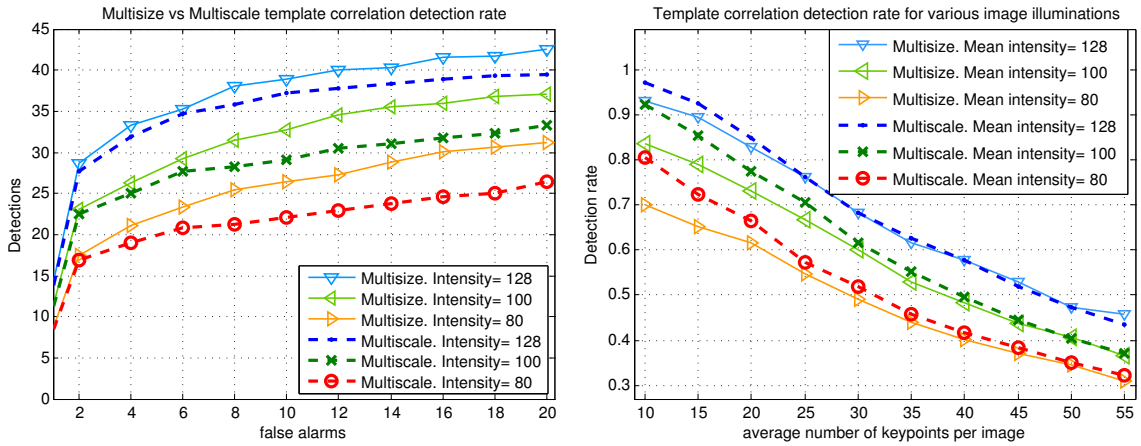


Figure 4.11: Comparison of the detection rate using multiscale template matching or multisize. Results are shown for various image mean intensity levels

image projection can be expressed as

$$\begin{pmatrix} su_i \\ sv_i \\ s \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{4.10}$$

where $m_{11}, \ldots, m_{34}$ are the projection parameters dependent on the stereo rig calibration, and s is an scale factor. These 2D projections are the two points on right and left image which form the feature correspondence pair. Let $\mathbf{x}_{i,}^r$ and $\mathbf{x}_{i,}^l$ be the 2D projections of the feature i on I_0^r and I_0^l. Using the relation between a 2D projection and $\mathbf{X}_i$ stated on equation (4.10), we can write $\mathbf{X}_i$ as

$$\begin{pmatrix} f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{11} & f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{12} & f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{13} \\ f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{21} & f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{22} & f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{23} \\ f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{31} & f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{32} & f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{33} \\ f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{41} & f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{42} & f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)_{43} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \rightarrow AP = b, \tag{4.11}$$

where $f(\mathbf{x}_{i,}^r, \mathbf{x}_{i,}^l, p)$ is a function of $m_{11}, \ldots, m_{34}$, the camera parameters and $\mathbf{x}_i^{\{r,l\}}$. The 3D coordinates $\mathbf{X}_i$ can be computed minimising the system on equation (4.11).

Minimisation is applied to all the feature correspondences, obtaining a initial set of n' 3D points

$$\{\mathbf{X}_i\}_{i=1\ldots n'}. \tag{4.12}$$

From the initial set of n' points, some correspondences may be false alarms, that is, erroneous matched interest points, and must be filtered out before generating the model.

The filtering process takes into account face geometrical constraints, like shape and position to ensure the rejection of points outside the face bounds.

4.2.1 Cylinder model fitting and feature self-occlusion

One of the common problems to face pose estimation systems based on tracking methods, as stated on chapter 2, is the self occlusion. The face model features can self-occlude when the head turns over a certain angle, so some of the model points may not be visible. To

detect these features in advance, a hidden-point pattern is created during model initialisation. Each feature is associated to two limit rotation angles within it is visible. When the face rotation angle is over the limit angles of a point, it is considered to be hidden and it is not used for tracking and pose estimation.

To create the hidden-point pattern, a vertical-oriented cylinder is adjusted to the $\{\mathbf{X}_i\}_{i=1\ldots n'}$ feature coordinates [Eberly 03], as shown in figure 4.12. The minimisation is implemented inside a RANSAC loop to avoid fitting the cylinder to the most salient points, such as those of the nose. The outliers threshold h_{Cyl_RANSAC} is chosen small enough so that nose points are outliers to the initial minimisation. On each RANSAC iteration t, a group $\mathcal{N}^{(t)}$ of seven random points is generated, and a cylinder is adjusted to minimise the error function

$$\mathcal{E}(\theta^{(t)}) = \min_{\theta} \sum_{k \in \mathcal{N}^{(t)}} \mathcal{E}_k(\theta), \tag{4.13}$$

where

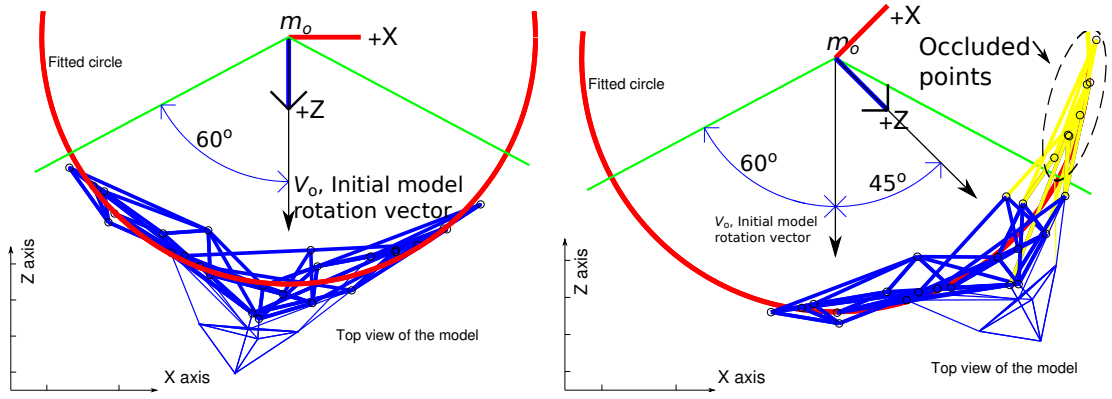
$$\mathcal{E}_k(\theta) = \sqrt{(x - x_k)^2 + (z - z_k)^2} - r^2 \tag{4.14}$$

is the individual 3D point error function and $\theta = (x, z, r)$ is the parameter list in the minimisation, and the parameters represent the centre in the (X, Y) plane and the radius of the fitted cylinder. After each iteration, inliers are calculated as

$$\mathcal{I}^{(t)} = \{\mathbf{X}_i\} : \mathcal{E}_i(\theta^{(t)}) < h_{Cyl\_RANSAC}, \quad i = 1 \ldots n', \tag{4.15}$$

and the best iteration is chosen to maximise the number of inliers. After the RANSAC has found the largest set of inliers $\mathcal{I}$, a new minimisation is executed using all these inliers to find the best set of parameters $\theta_o = (x_0, z_0, r_0)$:

$$\theta_o = \arg\min_{\theta} \sum_{k \in \mathcal{I}} \mathcal{E}_k(\theta). \tag{4.16}$$



(a) Frontal face (b) Partly occluded face

Figure 4.12: Circle fitted to the face to get the limit angles

After minimising (4.16), the point $m_o = (x_o, z_o)$ is defined as the model geometric centre. All the angles have an offset inherited from the initial model rotation offset. Each point of the model is considered hidden when its angle with respect to the initial model rotation vector $\vec{V}_0$ exceeds $\pm 60°$ degrees.

After computing the model geometric centre m_o, those 3D points for which the distance d_i to m_o is outside a given range are rejected, i.e., the points are far from the 3D cylinder surface, typically 50mm $< d_i <$ 120mm, since these points are probably outliers.

The self-occluding model of the face has several advantages. Although the exact occluding angle for each feature is not known, since detailed geometry of the face is not computed, the self-occluding model gives a prediction about when this is likely to happen. This prediction allows to reduce the number of erroneous feature matches at the tracking stage caused by features that are occluded.

Figure 4.13 shows the appearance similarity variation of a feature texture over rotation compared to its frontal, initial appearance. Zero-mean normalised cross-correlation was used to compare the patches. It shows how correlation similarity decrease when the rotation angle increases. Although a new re-registering technique will be introduced in the next chapter to overcome this issue, we choose to discard features for pose estimation when their similarity is likely to be low, in order to reduce the probability of tracking errors.

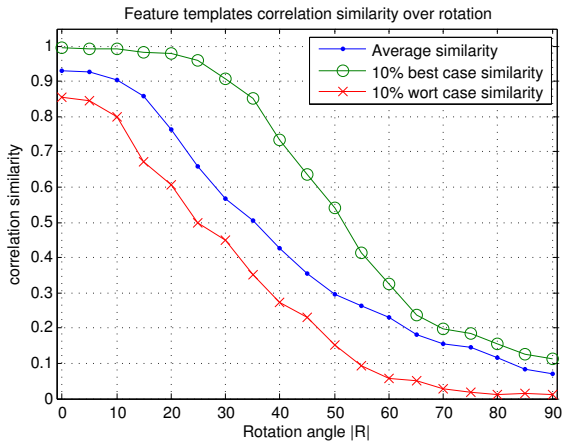


Figure 4.13: Feature appearance similarity for different face rotations. The plot shows how similar is the appearance of a feature under different view-points, for the average case, or taking into account the 10% best and worst cases

Finally, the self-occluding model also reduces the computational cost, since all the features that are occluded are not processed.

4.2.2 Model formation

Initially, n' correspondences were extracted, of which only a set of N_0 are correct and used to form the model $\mathcal{M}$. The model centre is set to $m_o = (x_o, y_o, z_o)$, where (x_o, z_o) are obtained from the cylindrical model fitting, and y_o is set to

$$y_o = \frac{1}{N_0} \sum_{i=0}^{i<n', i \in \mathcal{M}} y_i \tag{4.17}$$

The correct correspondences are sorted and translated from camera coordinate frame to object coordinate frame reference system to form the model.

$$\mathbf{X}_i^{(\mathcal{M})} = \mathbf{X}_i - m_o \tag{4.18}$$

Each model feature i is formed by the 3D point coordinate $\mathbf{X}_i^{(\mathcal{M})}$ and a cluster $\mathbf{C}_i$ of appearance descriptors obtained from the 2D location of the interest points from which the feature was extracted on each camera. Since correlation is being used for feature tracking, following discussion in section 4.1.2, the appearance descriptors which form the clusters are the patches captured from the images, and will be called *textures* hereinafter, denoted as $\{\mathbf{T}_i^{(r)}, \mathbf{T}_i^{(l)}\}$. These, however, are not the only features which form the model, since it expects new ones, up to N, to be added during tracking to reveal parts of the face initially occluded. Therefore, the model is formed as

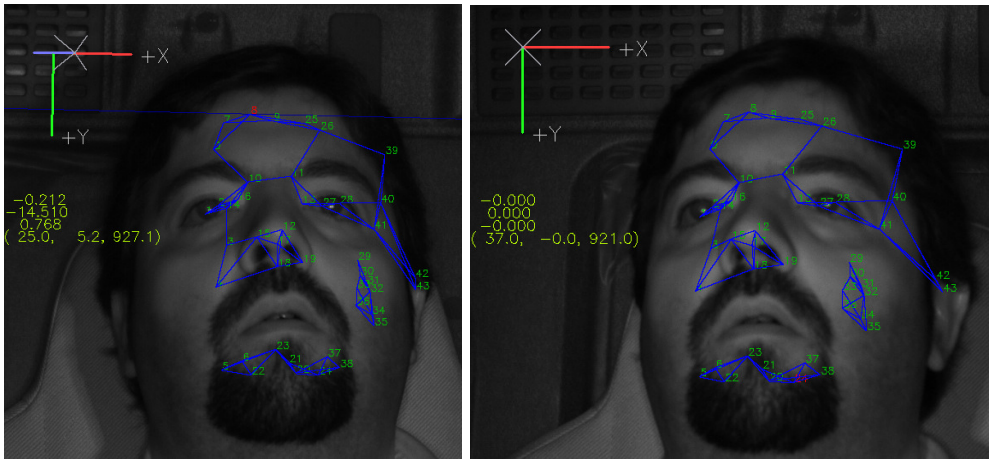
$$\mathbf{C}_i = \{\mathbf{T}_i^{(r)}, \mathbf{T}_i^{(l)}\}, \tag{4.19}$$

$$\mathcal{M} = \{\mathbf{X}_i^{(\mathcal{M})} + \mathbf{C}_i\}_{i=1\ldots N_0,\ldots N}, \tag{4.20}$$

where $\mathcal{M}$ is the 3D face model and $\mathbf{X}_i^{(\mathcal{M})}$ are the 3D points of the model in the object coordinate frame. Initial object pointing vector is defined as $\vec{V}_0 = (0, 0, 0)$, with origin in the model centre m_0, and referenced to the right camera frame system.

The coordinates of the model $\{\mathbf{X}_i^{(\mathcal{M})}\}$ are initially set rigidly, and distance between them is constant, as represented by the blue lines in figure 4.14. However, methods to dynamically adjust $\{\mathbf{X}_i^{(\mathcal{M})}\}$ and $\{\mathbf{C}_i\}$, that is, model structure and appearance, and to extend the model will be presented in chapter 5.

Figure 4.14 depicts the projections $\mathbf{x}_i^{\{r,l\}}$ over the camera images I_0^r and I_0^l respectively of the model points $\mathbf{X}_i^{(\mathcal{M})}$. Projection points are calculated applying equation (4.10). Figure 4.15 shows and example of an automatically generated model from the images showed on figure 4.14. Each vertex is a 3D model point, $\mathbf{X}_i^{(\mathcal{M})}$.



(a) Set of projections $\{\mathbf{x}_{i,}^l\}$ over I_0^l (b) Set of projections $\{\mathbf{x}_{i,}^r\}$ over I_0^r

Figure 4.14: Projections $\mathbf{x}_{i,}^l$ and $\mathbf{x}_{i,}^r$ of model points over the left and right images, I_l and I_r

4.3 Conclusions

The main purpose of the 3D model is to provide a reference from which the pose can be extracted from 2D feature projections on the camera images. However, it can also help to detect and correct tracking errors and feature self-occlusions.

This chapter proposes a method to automatically generate a 3D coarse face model formed by 3D points and their appearance descriptors in the images using a stereo rig. This

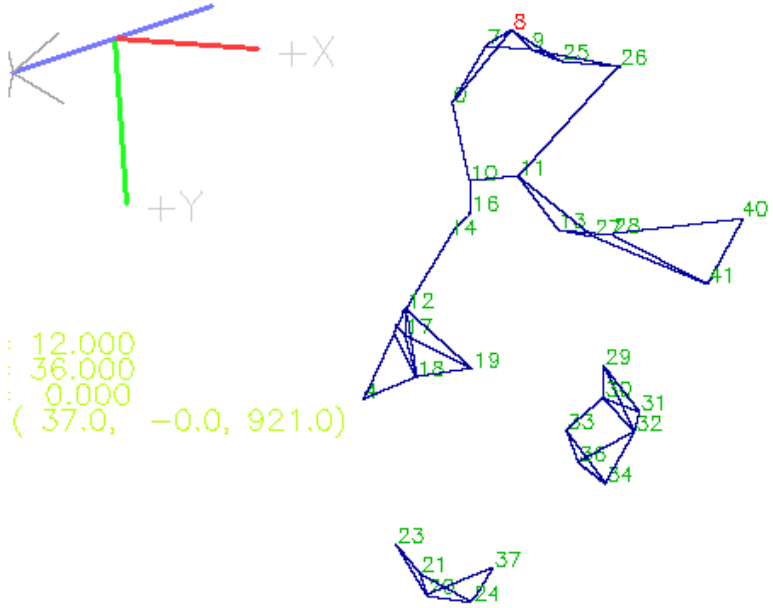


Figure 4.15: 3D face model showing the set of points, $\mathcal{M} = \{\mathbf{X}_i^{(\mathcal{M})}\}_{i=1...N}$

is done by automatically extracting and matching face features from both camera images. No matter the matching method being used, including incorrectly matched features in the model is an issue that must always be addressed. For this purpose, an improved matching scheme has been proposed, based on matching feature patches of different sizes. This technique takes advantage of the precision provided by smaller patches, and the robustness of using bigger ones.

To deal with self-occlusions, we have defined a self-occlusion limit angle for the model points. This provides an *a priori* knowledge of which points are not visible, and also allows to reject those features that might appear too distorted to be correctly tracked when the face has an angle close to a feature self-occlusion angle.

Chapter 5

Face Pose Estimation with Model Corrections

This chapter presents the frame to frame execution of the algorithm, which involves the face tracking and pose estimation processes. After an initial face model has been created as explained in chapter 4, the algorithm described in this chapter executes in a loop frame to frame to achieve the final goal: to accurately determine the pose of the face on each frame. The steps involved in this task are described in the next sections. Figure 5.1 depicts a flow chart of the execution of the different processes.

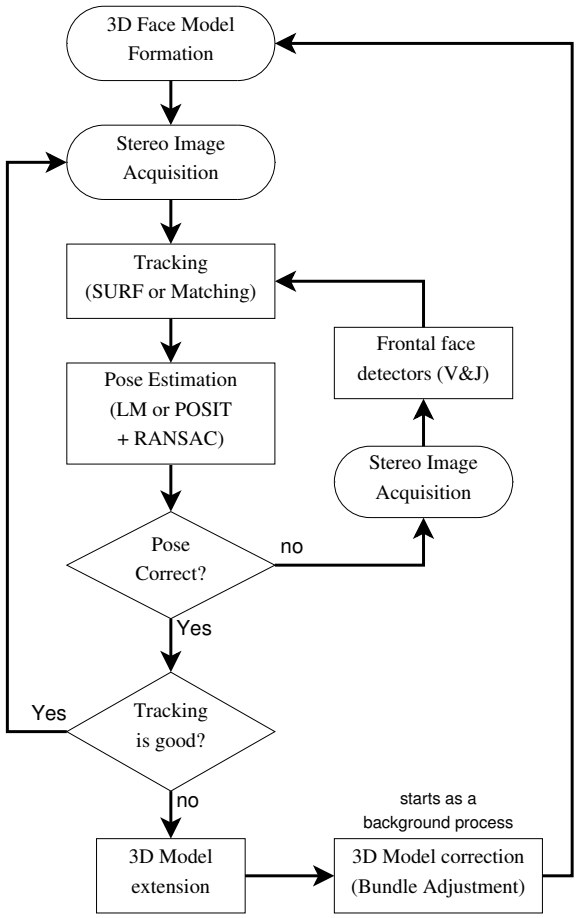


Figure 5.1: Schematic flow chart of the face tracking and pose estimation algorithm

5.1 Feature Tracking

As we described in chapter 3, the next step after the model creation is the frame to frame tracking of the feature set which forms the model. This tracking allows fitting the model to the face in the frame being processed. Since the appearance of the features can drastically change as the head rotates or due to illumination, the tracking technique to be used must be capable of dynamically updating the features descriptors to account for these changes. However, any dynamic updating process also involve the possibility of drifting due to the accumulation of small errors over time. To address this problem, the tracking is supported on the 3D face model, to provide a correction method for drifting points.

The reasoning behind the usage of the 3D face model as a correction mechanism is that obtaining an *a priori* appearance model is a difficult task: it requires comprehensive image databases of enough objects under different deformations (and sometimes illumination conditions), hand or semi-automated marking of the images, and may also involve post-processing. Obviously, a different model needs to be created for each kind of object. Consequently, if *a priori* information on the features can not be used, the only initial information available for tracking is the first frame appearance and the 3D model itself.

Several methods that work without *a priori* information have been presented in the literature. Most of them have focused on tracking image descriptors on a video sequence. The classic approach is to use patches extracted on the first frame of the sequence to search for similarities on the following ones. Lukas-Kanade method [Lucas 81] was one of the first proposed solutions and it is still frequently used. This algorithm uses Newton's optimisation method to find the best matching patch. More recent approaches include works that relay in more complex modelling of the image descriptor to increase the robustness and precision of the tracking. Jepson *et al.* [Jepson 03] presented a system with appearance based on three components: a stable component that is learnt over a long period based on wavelets, a 2-frame tracker and an outlier rejection process. This method works robustly for large patches, but not so well for smaller ones. Its computational requirements are also high, and it does not work in real time. [Moreels 07] showed that the repeatability performance of this State of the art detectors slowly degrades with increasing change of viewpoint. Typically, face features are not highly contrasted or textured, in contrast with those used in other 3D structure from motion applications such as [Davison 07]. A face does not typically present corners, and it is exposed to illumination variations and shines. Moreover, the limited illumination leads to dark images, which makes features even less contrasted. Another important issue derived from low illumination conditions are the poorly focused images. The iris of the camera must be wide open to allow the maximum light into the sensor, drastically reducing the depth of field. This produces that some parts of the face, specially if the driver moves forward or backwards, might not be well focused.

Several other techniques can be used for tracking. Yin and Collins [Yin 07] made a successful approach to object tracking without *a priori* information by means of template correlation on the camera 2D projection space. The patches were selected using a Harris Corner detector. Yin and Collins showed that template correlation techniques with Harris corners are able to extract and match more features than descriptor matching algorithms, such as *SURF* or *SIFT*. Moreover, the detection step is only needed once at the initialisation step.

Four different approaches were studied on section 4.1 for the process of finding the correspondences across camera projections: SURF, simple patch correlation, multiscale

correlation and multisize matching. The schemes tested and used for frame to frame feature tracking are the same.

In addition to the discussion held in the previous chapter, the techniques used now for tracking have the requirement of real-time performance. Moreover, since a feature descriptor is stored at the first frame and tracked along face rotations, the rotation range is now much wider than in the stereo matching stage. SURF proved very bad results due the low repeatability rate caused by the difficult lighting conditions, while in addition is the slowest of all the mentioned techniques. On the other hand, the multiscale scheme did not perform as expected either. Due to the much wider rotation range, there is a higher uncertainty on the scale on which it is going to have better performance after some wide rotation. Moreover, the smallest scales have an intrinsic localisation error due to the low resolution of the scale where the feature is located.

For these reasons, and for simplicity, the matching technique that we have used for frame to frame feature tracking is the same used for stereo matching: the multisize patch correlation, explained in section 4.1.2. The only difference is a more restrictive correlation threshold to minimise tracking error and outliers. In equation (4.9), h_{tc} is now set to 0.7.

However, as the threshold h_{tc} is increased, tracking of a feature fails at smaller rotations, when its dissimilarity to the original stored texture is big enough. A smaller threshold, on the other hand, would increase the tracking errors and the number of outliers to the subsequent pose estimation step, reducing the algorithm capability to detect and reject these outliers. As an attempt to solve this problem, we have tested a feature template warping technique.

5.1.1 Warping of feature points projections

The main problem to solve when tracking 3D objects is the changing appearance of the feature points as its pose changes. To deal with this problem, many authors apply a patch warping technique [Dornaika 04, La Cascia 00, Xiao 02]. Consider a feature texture that has been first captured and stored with a face pose $\mathbf{P}_0$. For any given pose of the face $\mathbf{P}_1$, other than the initial pose $\mathbf{P}_0$, the feature has a different projection angle. This makes it maybe have a very different appearance than that initially captured with $\mathbf{P}_0$. If the 3D surface of the feature patch were known, its projection could be *warped* or mapped from the patch with view-point $\mathbf{P}_0$ into the patch with view-point $\mathbf{P}_1$ as a piece-wise affine transformation [Baker 04a], $\mathcal{W}$. Formally, the warping process applied to an input patch I_0 captured with pose $\mathbf{P}_0$ is denoted

$$I_1 \approx I_0' = \mathcal{W}(I_0, b), \tag{5.1}$$

where I_1 is the patch with view-point $\mathbf{P}_1$; I_0' is the approximation to I_1 obtained from warping I_0 as it would be seen from a point of view with pose $\mathbf{P}_1$; and b denotes the geometrical parameters of the transformation from $\mathbf{P}_0$ to $\mathbf{P}_1$. It is necessary to find the transformation that makes I_0' as similar as possible to I_1.

Three classes of warping have been considered. The most generic case is when the patch to be warped represents the projection of a generic 3D surface. In this case, an arbitrary warping is needed, generally applied by splines defining the curvature of the 3D surface. However, the 3D surface of the face is roughly represented by the model, with only a few values at the positions where each feature has been detected, but not in a near neighbourhood around each feature itself. This class of warping could be applied if a dense matching of the face is performed and the 3D surface around each feature is known. If this is not the case, a feature patch can be assimilated to a planar surface if the patch

is small enough. Two classes of warping are possible for planar surfaces. A more realistic case is the projective warping, which emulates a camera projective model. If the scale of the patch compared to the distance to the camera is small enough, this can be simplified to an affine warping, which assumes an orthogonal projection model. Figure 5.2 depicts the three warping approaches.

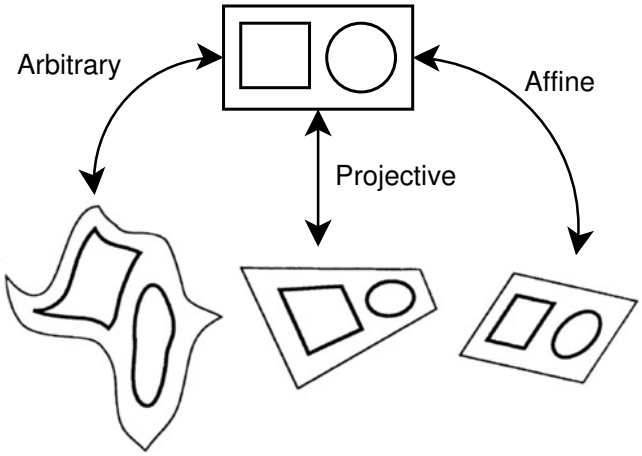


Figure 5.2: Warping classes

We use an affine warping model for tracking face features since patches are very small to have any noticeable perspective deformation because the scale of the scene compared to the distance to the camera is small. Moreover, the errors caused by assuming a planar surface are far greater than those from assuming an orthogonal camera model. In an affine transformation, straight lines remain straight and parallel lines remain parallel, but distances and the angles between intersecting lines may change. Equation (5.1) can be expressed as an inverse pixel mapping relation $\mathbf{x}' \rightarrow \mathbf{x}$, which maps any pixel in the destination patch I'_0 from subpixel positions in the source I_0 as

$$\mathbf{x} = \mathcal{W}^{-1}(\mathbf{x}', b) \tag{5.2}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = W \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}, \tag{5.3}$$

where W is the inverse warping matrix. Equation (5.2) maps any integer pixel $\mathbf{x}' = (u', v')^T$ in I'_0 from $\mathbf{x} = (u, v)^T$ in I_0, and can be calculated as

$$\begin{aligned} u &= m_{00} * u' + m_{01} * v' + m_{02} \\ v &= m_{10} * u' + m_{11} * v' + m_{12}. \end{aligned} \tag{5.4}$$

In equation (5.4), the resulting source image pixel $\mathbf{x} = (u, v)^T$ points to a non-integer pixel position. The intensity value for that position is computed using an interpolation method, such as bilinear or bicubic from the surrounding pixels within I_0.

A transformation matrix W_i has to be inferred for each feature i of the model from the limited knowledge of the 3D structure of the face. For a given point of the model, $\mathbf{X}_i = (x_i, y_i, z_i)$, we do not know how the 3D surface is around the feature. What we know is the position of surrounding features, which gives a rough idea of the orientation of the surface $\mathbf{S}_i \in \mathbb{R}^3$, assumed planar and centred in the 3D point $\mathbf{X}_i$. Let $\mathbf{X}_j$, and $\mathbf{X}_k$ be the two 3D face model points closest to $\mathbf{X}_i$. The set $(\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k)$, with known

projections $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ over the camera image, form a triangle in $\mathbb{R}^3$, which is enough to calculate the coarse orientation of the surface $\mathbf{S}_i$. The normal vector to $\mathbf{S}_i$ is know as the *pseudo-normal* vector to $\mathbf{X}_i$, i.e., it represents an approximation of the normal pointing vector of a 3D surface centred at $\mathbf{X}_i$. When the face moves to pose $\mathbf{P}_1$ the set of points $(\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k)$ moves to $(\mathbf{X}_i^1, \mathbf{X}_j^1, \mathbf{X}_k^1)$, $\mathbf{S}_i$ moves to $\mathbf{S}_i^1$, projection points over the camera image move to $(\mathbf{x}_i^1, \mathbf{x}_j^1, \mathbf{x}_k^1)$, the patch associated with the feature changes from P_i to P_i^1.

The objective of the warping is to find the transformation $P_i \rightarrow P_i^w$ which minimises $P_i^w - P_i^1$, assuming $\mathbf{S}_i$ is planar. To calculate the transformation matrix $W_i^{\mathbf{P}_1}$ which transforms P_i to P_i^w, it is necessary to obtain the new positions of $(\mathbf{X}_i^1, \mathbf{X}_j^1, \mathbf{X}_k^1)$ and its corresponding 2D projections $(\mathbf{x}_i^1, \mathbf{x}_j^1, \mathbf{x}_k^1)$ over the camera image. Now we can set $\mathbf{x}'$ from equation (5.2) as $\mathbf{x}_i^1$, $\mathbf{x}_j^1$ and $\mathbf{x}_k^1$ in a system of three equations. Equation (5.4) is applied to the three points together to form a system of the form $A \cdot M_i^{\mathbf{P}_1} = B$ which can be expanded as follows to solve $W_i^{\mathbf{P}_1}$:

$$\begin{pmatrix} u_i' & v_i' & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_i' & v_i' & 1 \\ u_j' & v_j' & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_j' & v_j' & 1 \\ u_k' & v_k' & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_k' & v_k' & 1 \end{pmatrix} \cdot \begin{pmatrix} m_{i,00} \\ m_{i,01} \\ m_{i,02} \\ m_{i,10} \\ m_{i,11} \\ m_{i,12} \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{pmatrix} . \tag{5.5}$$

Figure 5.3 shows the projection of feature points and the patch transformation.

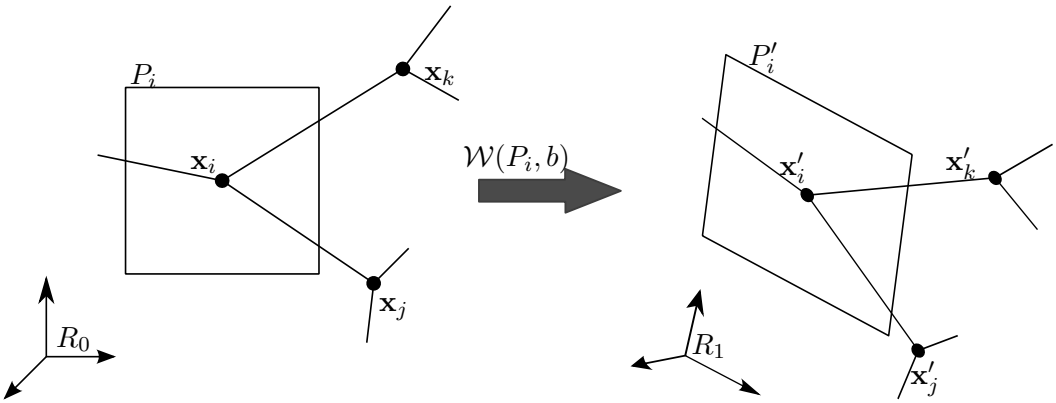


Figure 5.3: Warping process of a feature i. View in $\mathbb{R}^2$

Similarly, the same process, using four points instead of three could be inferred if a perspective warping were to be applied. However, the approach above is based on the assumption that the *pseudo-normal* to the surface $\mathbf{S}_i$ at model points $\mathbf{X}_i$ is correctly computed. The closer the other chosen points used to calculate $\mathbf{S}_i$ are to $\mathbf{X}_i$, the better the approximation would be, but since a face is an irregular surface and the 3D model is not a dense structure, there is some error on the calculation of $\mathbf{S}_i$. Most of the times, this error is much bigger than that derived from using affine transformation instead of projective. Consequently, for simplicity and time performance, applying a perspective transformation is not advisable.

In practise, we apply the template warping between a stored model texture of a feature $\mathbf{T}_i$, and the feature patch at the current frame, $P_{i,t}$. To calculate W_i the pose $\mathbf{P}_t$ has to be known at frame t, but it is unknown because it is calculated after the tracking step. We can, therefore, either predict it using techniques such as a Kalman filter, or use $\mathbf{P}_{t-1}$

instead, assuming $\Delta\mathbf{P}$ is small enough since the system is working at 30 fps. Furthermore, warping can be applied in any of two directions: $\mathbf{T}_i \rightarrow P_{i,t}$ or $P_{i,t} \rightarrow \mathbf{T}_i$. In the first case, the stored $\mathbf{T}_i$ is warped to create $\mathbf{T}'_i$, and this is compared to $P_{i,t}$ at the correlation stage. In the later case, the current feature patch on the image, $P_{i,t}$ is warped to create $P'_{i,t}$, and compared to $\mathbf{T}_i$.

The factors which determine how to do the warping are the feature localisation accuracy after the matching stage, and the real time performance. An exhaustive testing have been carried out. To obtain the error in feature localisation, $\mathbf{P}_t$ is extracted from the ground-truth (GT), and the real feature location calculated projecting the feature 3D points using the GT pose. The ground-truth is explained in details in the results chapter, section 7.2. The warping transformation matrix W_i is also computed using the data calculated from the GT, to isolate the warping error from feature localisation error. This assumes that the feature input position is correct for the comparison.

The curves in figure 5.4 shows the error for three of the possible warping combinations. It can be observed how there is very little different between them. Consequently, for simplicity the warping approach applied in the proposed system in this thesis is $\mathbf{T}_i \rightarrow P_{i,t}$. Figures 5.5 and 5.6 show some examples of warping for various face features under different rotations. The first figure shows the results applying the warping $\mathbf{T}_i \rightarrow P_{i,t}$, and the second applying $P_{i,t} \rightarrow \mathbf{T}_i$.

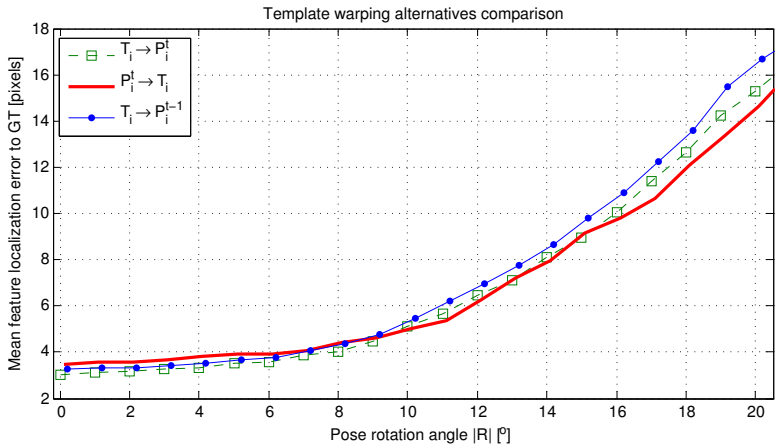


Figure 5.4: Comparison of localisation error for various template warping alternatives

Due to pseudo-normals calculation errors and feature not being planar surfaces, the template warping works reliably for small rotations. However, it does not solve the feature dissimilarity problem produced for wide rotations. To deal with this, we propose a technique to capture or *re-register* feature textures as the head rotates. The re-registering happens at some certain frames at which the uncertainty of the pose estimation is likely to be very low. This allows us to *re-register* the new appearance of the features as the face rotates, so the textures used for correlation are not always the ones captured at initialisation. This reduces the dissimilarity while using a high threshold h_{tc} for the matching as we indicated in equation (4.9).

5.2 Feature re-registering

As the head rotates, feature appearance changes to levels at which it is impossible to establish a correspondence using any matching algorithm over the initially registered

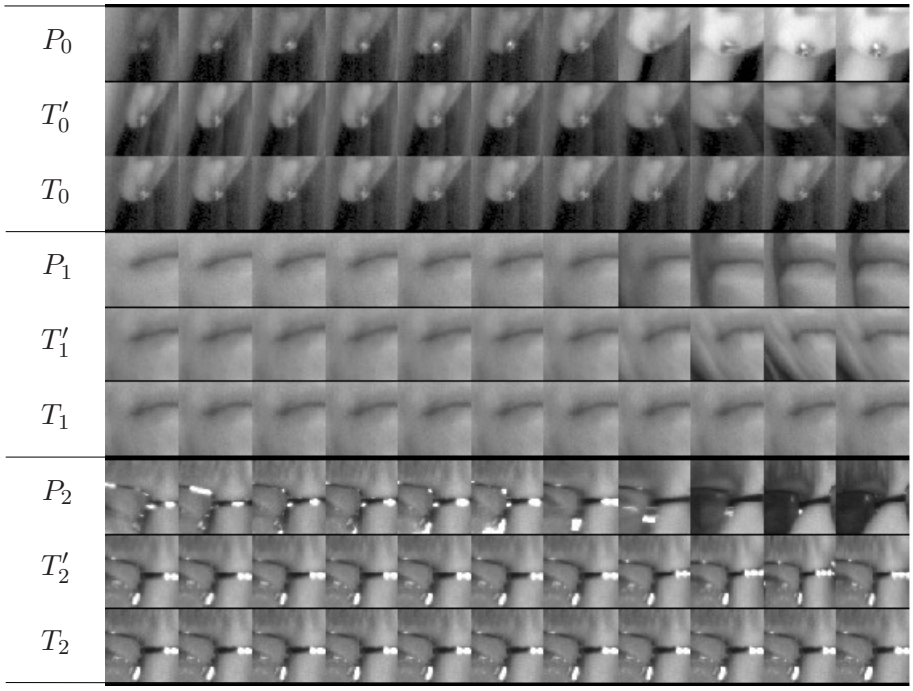


Figure 5.5: Warping for various face features under different small rotation angles. For each feature i there are three rows. The first row depicts the patches on the camera image, $P_{i,t}$, under different rotations. The last row shows the unwarped texture of the feature, $\mathbf{T}_i$, as it is stored in the model. The second row shows the stored texture in the third row, warped to simulate the first one, $\mathbf{T}'_i = \mathbf{T}_i \rightarrow P_{i,t}$

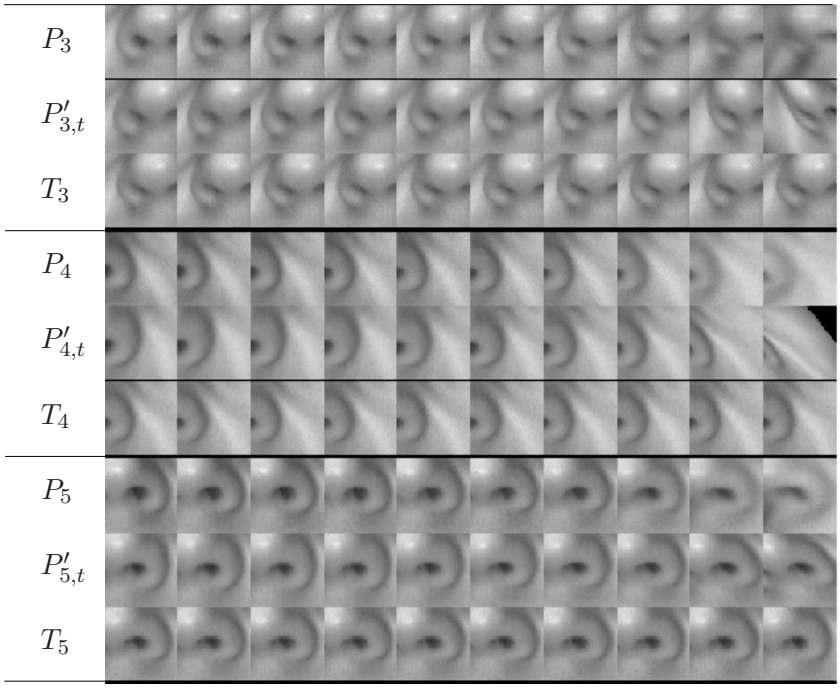


Figure 5.6: Warping for various face features under different small rotation angles. Now, the inverse wrapping have been tested, and the middle row for each feature depicts $P'_{i,t} = P_{i,t} \rightarrow \mathbf{T}_i$

feature textures. Some algorithms proved to deal better with rotations than others, but none is capable of finding the correspondences under wide rotations if no extra information is provided by other means. As face features are not planar in shape, in general it is not a good solution to try a template warping to correct changes in appearance. Moreover, this process is costly, and often needs some *a priori* information about the orientation of the patch in the 3D space.

To deal with this appearance variation we have developed a new *re-registering* technique based on using different view-angles of the face from the two cameras to try to estimate the appearance that a feature will have after some rotation. The main idea behind our re-registering technique is to capture new textures from feature patches and to store them in the model when we know that the pose estimation error is the lowest possible. At the model creation step, images patches with different view-points are captured from both cameras, and stored in the model. Instead of using disjointed appearance models for each camera, the stored textures are grouped together for each feature in a cluster. At the tracking stage, some elements of the cluster are correlated in the image, and the one giving better correlation results is used for feature localisation [Nuevo 10].

Figure 5.7 makes a comparison of the mean feature localisation error with one camera using for matching the initial feature patches, stored in the model as feature textures. One curve shows the error using for correlation only the initial texture captured with this same camera. The other curve shows the error if we take the best result of the correlation of the textures initially captured on the each of the cameras. It shows how localisation error is drastically reduced using for tracking a cluster containing the stored texture captured from both cameras . The dashed vertical line shows the angle between the two cameras from a distance of approximately 90cm, at which the face is usually located. Localisation error has a minimum precisely at these rotation angle, because the view-point of the face from one camera is the same as the view-point from the other after a rotation of approximately 15°.

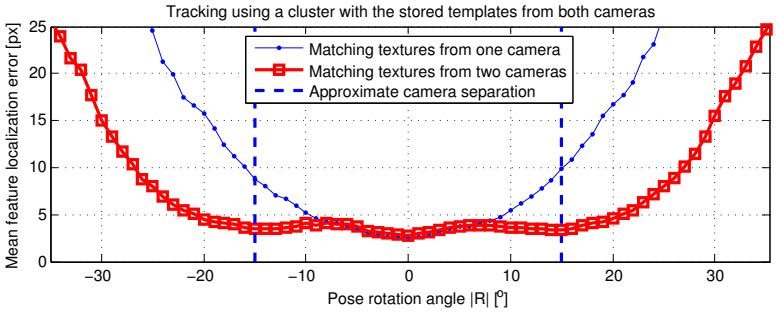


Figure 5.7: Comparison of localisation error using the textures from one or the two cameras in a cluster

Figure 5.7 shows that the optimal angle to perform re-registering is equivalent to the camera separation, because with this yaw rotation the localisation error is minimal. At these rotations, new textures from the feature patches on the image can be captured and stored in the model, since it is at this rotation when the localisation error of these patches is likely to be minimal. Repeating this process all over the yaw rotation range, tracking error can be kept very low under full range rotations.

Following this scheme, new features appearances are stored in clusters at certain angles α_j, from both camera frames. Let $\mathbf{T}_i^{(j)}$ be a stored texture for feature i and view-point angle α_j, no matter from which camera it was captured. For each feature point belonging

to the model, a cluster of textures $\mathbf{C}_i$ is also stored along with $\mathbf{X}_i$ with feature textures from different angles α_j,

$$\mathbf{C}_i = \{(\mathbf{T}_i^{(j)}, \theta_i^{(j)})\}, j = \pm 1, j = \pm 2, \ldots. \tag{5.6}$$

The texture $\mathbf{T}_i^{(t)}$ used for correlation at a certain frame t to search for the feature location in the tracking process is

$$\mathbf{T}_i^{(t)} = \arg\min_j (\mathbf{T}_i^{(j)} - P_{i,t}). \tag{5.7}$$

Let $\mathbf{P}_0$ be the 3D model pose at $t = 0$. From this pose, a model point $\mathbf{X}_i$ projects with view-point angle α_0 at $\mathbf{x}_{i,0}^r$ on image I_0^r. $P_{i,0}^r$ is the patch around this point seen from camera C_r. We assume that $\mathbf{P}_0$ is correct by definition, since it is the initial 3D model pose. The texture $\mathbf{T}_i^{(0)} = P_{i,0}^r$ from feature i is stored to the cluster, which is also correct since we are at the initialisation, i.e., has no drifting. Similarly, β_0 is the projection angle of the feature to the left camera, and $\mathbf{T}_i^{(1)} = P_{i,0}^l$ from image I_0^l is also stored to the cluster. This process can be followed in figure 5.8 ①, which illustrates the re-registering mechanism.

Now, let the face rotate to its left for a certain time after initialisation and model creation have finished. Let $\mathbf{P}_1$ be the pose at a time t_1 for which the projection angle, β_1, of model point $\mathbf{X}_i$ into the other camera image $I_{t_1}^l$ is similar to α_0, or more precisely, lower than an error threshold

$$\epsilon > \beta_1 - \alpha_0. \tag{5.8}$$

If such is the case, the patch $P_{i,1}^l$ should be very similar to the stored $\mathbf{T}_i^{(0)}$, previously captured from the other camera, since the projection angles to the respective cameras are the same. Thus, now $\mathbf{T}_i^{(0)}$ can be used to track the new position of $\mathbf{x}_{i,1}^l$ on image $I_{t_1}^l$ more accurately since $\mathbf{T}_i^{(0)}$ has the same view-point than $P_{i,1}^l$, and we previously assumed that this texture is correct. (Figure 5.8 ②)

At this time, the localisation error is expected to be minimal, and consequently it is convenient to re-register the texture of feature i. A new $\mathbf{T}_i^{(3)} = P_{i,1}^l$ is stored to the cluster, captured from the left camera, which should be very similar to $\mathbf{T}_i^{(0)}$, except for rotations in the roll and pitch angle and lighting conditions.

The angles α_j and β_j are not the same for all the features at a certain frame, so a new sub-index should be introduced, to denote the specific angle for each feature i at a time, $(\alpha_{j,i}, \beta_{j,i})$. However, in practise they are very similar since the size of the face compared with the distance to the camera is small. This means that the frame t_1 can be chosen so that condition in (5.8) are met for all the features,

$$t_1 : \sum_{i=0}^{i=N} |\beta_{t_1,i} - \alpha_{t_0,i}| < \epsilon'. \tag{5.9}$$

Error plot on figure 5.7 shows that there exists a pose $\mathbf{P}_1$ which satisfies this average minima in localisation error. A minima in average localisation error leads to a minima in pose estimation error, at the cost of a slight higher error in the captured $\mathbf{T}_i$, since t_1 does not minimise (5.8) for every single feature at a single frame, but minimise the sum of all of them. This condition implies that $\mathbf{P}_1$ error is also minimal at t_1.

Although the $\mathbf{P}_1$ error is not zero at frame t_1, it is minimum, so it is the best moment to register a texture from camera C_r. The 2D position $\mathbf{x}_{i,}^l$ of the feature is translated to

the right camera frame system, $\mathbf{x}^r_{i,}$, knowing $\mathbf{P}_1$. From this location in the right camera, another new texture $\mathbf{T}^{(2)}_i = P^r_{i,1}$ is also stored to the cluster.

Again, after some rotation, there is a time t_2 with pose $\mathbf{P}_2$ for which equation (5.9) is minimal,

$$t_2 : \sum_{i=0}^{i=N} |\beta_{t_2,i} - \alpha_{t_1,i}| < \epsilon', \tag{5.10}$$

and the last stored texture from camera C_r, $\mathbf{T}^{(2)}_i$ can be used to accurately search for $P^l_{i,2}$ in image $I^l_{t_2}$ on camera C_l (Figure 5.8 ③).

This process repeats over the whole yaw rotation range. If the face is rotating to its left, the camera C_l is mostly used for tracking and C_r to anticipate the view-point that C_l will have after a small further yaw rotation. Similarly, the process repeats in the opposite directions, when C_r mostly tracks and C_l anticipate the view-point of the features.

The described procedure generates a cluster as described in equation (5.6) of stored textures at discrete angles

$$\alpha_j \approx j \times \theta_c, \qquad j = 0, \pm 1, \pm 2, ..., \tag{5.11}$$

where θ_c is the average driver's view-point angle separation of the two cameras. For the stereo rig using is this thesis, $\theta_c \approx 15°$.

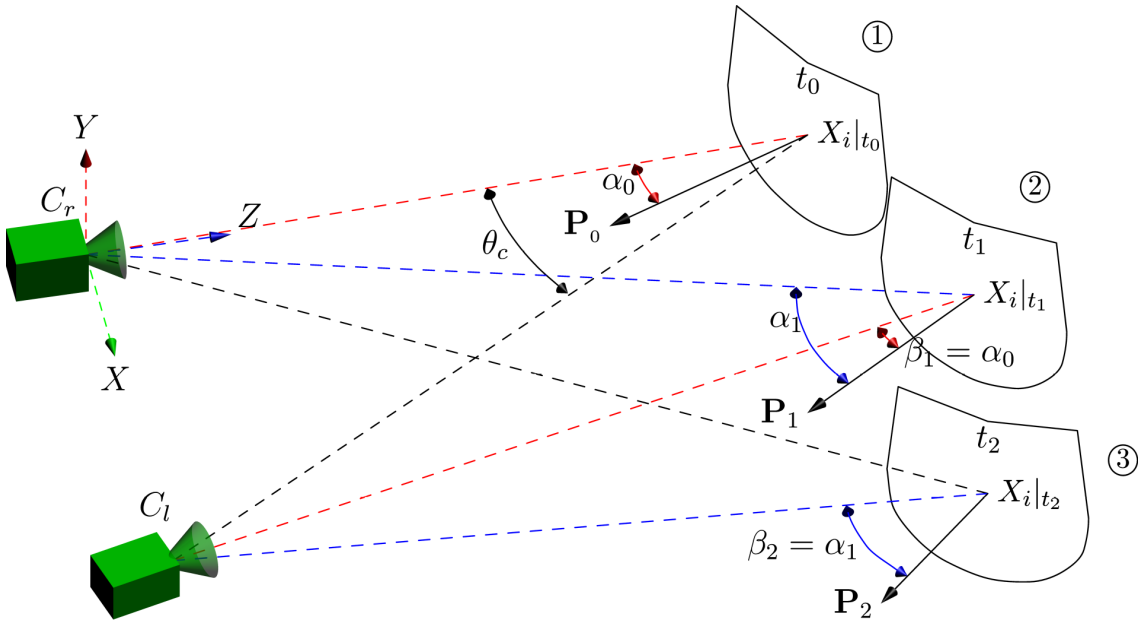


Figure 5.8: Re-registering process when the face is rotating to its left

Figure 5.9 shows the evolution of the correlation results for the tracking of $\mathbf{x}^r_{i,t}$ and $\mathbf{x}^l_{i,y}$ of a feature $\mathbf{X}_i$ over the right and left images when the face is rotating to the left. The graph shows the correlation peaks produced for $\mathbf{x}^r_{i,t}$ when re-registering takes place, at steps of approximately 15°. As for the graph of $\mathbf{x}^l_{i,t}$, its minimums represent the points at which the texture used for tracking switches from $\mathbf{T}^{(j)}_i$ to $\mathbf{T}^{(k)}_i, j \neq k$. This happens at ±7.5° from the re-registering rotations.

Similarly, if the face were rotated to the right, figure 5.9 would be symmetric, interchanging the functions acquired by the cameras.

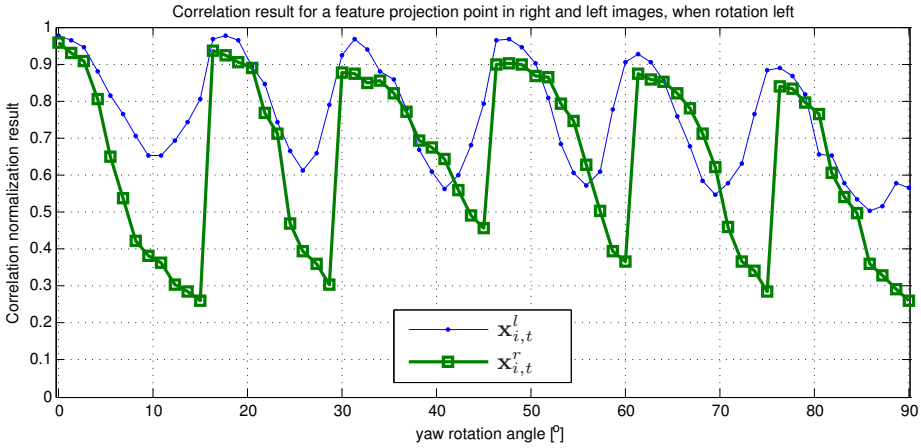


Figure 5.9: Correlation result of a feature patch for right and left images in a video sequence in which the face rotates to the left after initialisation. In the graph, times advance as the rotation angle increases. The peaks in the right image graph represent the moments at which re-registering happens

5.3 Pose Estimation

After the position of the tracking points have been updated for left and right frames, the 3D face pose has to be estimated using the putatives ($\mathbf{X}_i^{(\mathcal{M})} \Leftrightarrow \mathbf{x}_{i,t}^{\{r,l\}}$) of each feature, that is, the correspondences of the 3D points and their projections over one or both of the camera images.

Whether $\{\mathbf{x}^r_{i,t}\}$, $\{\mathbf{x}^l_{i,t}\}$ or both will be used to extract the pose depends on the pose estimation uncertainty for each frame and tracking error derived from the previous tracking step, as shown in graph 5.9. In the previous section, it was explained how the face is tracked mostly using only one of the cameras, depending if it is rotating left or right. If this is the case, the pose must be estimated based mainly in data from the frame which has been used for tracking. If the head is rotating left, then the tracking and subsequent pose estimation is performed over the left image. When the condition in equation (5.10) is met, the resulting pose is translated to the right camera, used to project the features 3D points over I^r_t to accurate obtain $\{\mathbf{x}^r_{i,t}\}$, and the textures from the patches around $\{\mathbf{x}^r_{i,t}\}$ in the image are re-registered. Similarly happens if the head is rotating right. If the head is moving randomly or it is static, pose is estimated from both frames, and the results averaged.

Two techniques to recover 3D face pose have been tested in this thesis, POSIT and Levenberg-Marquardt.

5.3.1 POSIT

Three-dimensional pose can be obtained using DeMenthon's four point iterative pose estimation algorithm (POSIT) [Dementhon 95]. The POSIT algorithm calculates the pose of a 3D rigid object from its 2D projection on a single image. It estimates the pose by first approximating the perspective projection as an scaled orthographic projection, and then iteratively refining the estimation until the distance between the projected points and the ones obtained with the estimated pose falls below an error threshold.

The pose $\mathbf{P} = \{R, T\}$ indicates the position of the central point of the model regarding to the camera coordinate system, and its rotation from the initial model given.

Let $\nu_t^{\{r,l\}}$ be the set of visible features each frame for the right and left cameras. The 3D face pose is computed individually for each camera frame in a RANSAC framework as

$$\mathbf{P}_t^r = f_{POSIT}(\{\mathbf{X}_k\}, \{\mathbf{x}_{k,t}^r\}), \quad k \in \nu_t^r, \tag{5.12}$$

$$\mathbf{P}_t^{l'} = f_{POSIT}(\{\mathbf{X}_k\}, \{\mathbf{x}_{k,t}^l\}), \quad k \in \nu_t^l, \tag{5.13}$$

where $\mathbf{P}_t^r$ is the pose estimation using the right camera, and $\mathbf{P}_t^{l'}$ is the estimation using the left one, and referred to the left camera frame system. This has to be translated to the right camera frame system, which is the reference frame,

$$\mathbf{P}_t^l = R_c \mathbf{P}_t^{l'} + T_c, \tag{5.14}$$

where $\{R_c, T_c\}$ are the rotation and translation from left to right camera.

The two poses, $\mathbf{P}_t^r$ and $\mathbf{P}_t^l$ are merged after the RANSAC process, depending on the output error of each estimation.

5.3.2 Levenberg-Marquardt algorithm

Levenberg-Marquardt (LM) algorithm [Marquardt 63] was first introduced by Donald W. Marquardt in 1963. It is an extension to the Gauss-Newton method, and can be interpreted as an intermediate method between Gauss-Newton and gradient descent. This addition makes Gauss-Newton more robust, meaning it can start far off the correct minimum and still find it. But if the initial guess is good, then it can actually be a slower way to find the correct pose. It works by dampening the parameter change that happens each iteration to make sure that Gauss-Newton always descends in parameter space.

Given the correspondences between 3D-points p_i , and its projections into a camera image at position p_i' (see 5.10). Pose estimation from these 2D-3D correspondences is about finding the rotation and translation between camera and object coordinate systems.

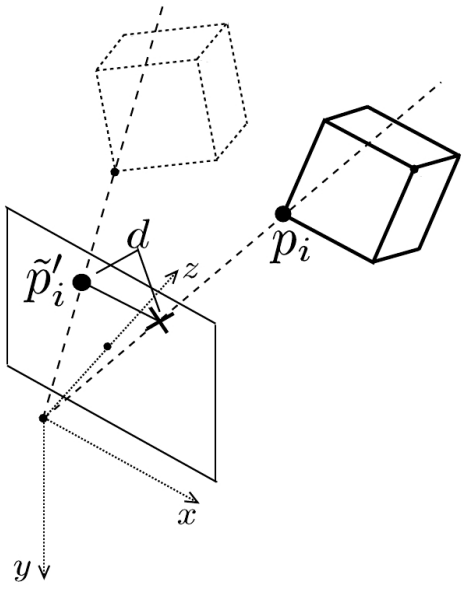


Figure 5.10: Geometric approach to pose estimation using LM.

LM estimates the relative rotation and translation of an object from an initial position and orientation (initial pose) to a new pose. The correspondences (p_i, p_i') are given for the

new pose. The algorithm can be formulated as a nonlinear least squares problem, which minimises the cost function

$$f_{LM} = \underset{\{R,T\}}{\arg\min} \sum_{k \in \nu^{\{r,l\}}} ||\mathbf{x}_k^{\{r,l\}} - proj(R\mathbf{X}_k + T)||^2, \tag{5.15}$$

where $proj()$ is the camera projection function and (R, T) are the rotation and translation matrices to be estimated.

The 3D face pose is computed individually for each camera frame in a RANSAC framework as

$$\mathbf{P}_t^r = f_{LM}(\{\mathbf{X}_k\}, \{\mathbf{x}_{k,t}^r\}), \quad k \in \nu_t^r, \tag{5.16}$$

$$\mathbf{P}_t^{l'} = f_{LM}(\{\mathbf{X}_k\}, \{\mathbf{x}_{k,t}^l\}), \quad k \in \nu_t^l, \tag{5.17}$$

As it is done when using POSIT, the two poses $\mathbf{P}_t^r$ and $\mathbf{P}_t^l$ are merged after the RANSAC process.

5.3.3 RANSAC

The matching process may not succeed for all points, and can result in errors or drifting for some of them. These errors negatively influence the accuracy of the estimated pose. Thus, a robust optimisation method is required to estimate the best fitting 3D face pose, that would detect as outliers the points that have been incorrectly tracked, so they can be safely discarded. The RANSAC algorithm is used to eliminate the outliers.

In each RANSAC iteration, seven points are randomly selected from the model, and used to calculate the pose (R and T matrices) using either POSIT or LM. With this R and T, all 3D visible points of the model, $\nu_t^{\{r,l\}}$, 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. RANSAC iterates until the reprojection error drops bellow 3 pixels, or until is has been iterating for approximately 15 ms, so real time performance is not compromised.

This process is performed over the frame used to track the points. In case both frames are used, the final pose estimation is calculated for each one, and the result given as the weighted sum, according to the next expressions:

$$R = \frac{R_r \cdot In_l}{In_l + In_r} + \frac{R_l^r \cdot In_r}{In_l + In_r}, \quad if\ In_r, In_l > In_{min} \tag{5.18}$$

$$T = \frac{\vec{T_r} \cdot In_l}{In_l + In_r} + \frac{T_l^r \cdot In_r}{In_l + In_r}, \quad if\ In_r, In_l > In_{min} \tag{5.19}$$

where In_l and In_r are the number of inliers from the left and right pose estimations, as determined with RANSAC. R and T are the resulting pose estimation. R_r and T_r are the pose estimation from I_r, and R_l^r and T_l^r are pose estimation from the left camera, translated to the right one using the corresponding stereo equations and calibration parameters. In case the number of inliers of any of the images is less the In_{min} threshold, that estimation is discarded and the estimation of the other camera is used.

Figure 5.11 depicts the effect of RANSAC, and compares POSIT and LM algorithms. The RANSAC error threshold is set to 35 pixels. The maximum feature localisation error in the curves not using RANSAC has been manually set to twice the RANSAC error

threshold, to 70 pixel. Other way, it is not possible to obtain any reasonable result for comparison because the tracking and pose estimation simply do not work. The graph compares the pose estimation error as a function of the feature localisation error, which is the output of the previous step.

LM algorithm is relatively stable for low to medium localisation error. For a mean localisation error of 20 pixels, the pose estimation error is still lower than 5°. Moreover, the presence of outliers —recall that the localisation error for the outliers is set to 70 pixels— do not degrade too much the pose estimation. POSIT, on the other hand, is more sensible to localisation error, and even more to outliers. To keep the POSIT measurements below an error of 5°, a localisation error no greater than 10 pixels is necessary. On the other hand, each LM execution needs as much as twice the time of a POSIT iteration. This means that during the approximately 15 ms that RANSAC is allowed to run, POSIT can perform much more iterations. In the even of many outliers, POSIT can detect and reject more outliers than LM, and so it deals a little better to occlusions than LM. If the number of outliers is low, POSIT gives more error than LM. In conclusion, LM is more accurate than POSIT, but the second is faster, allowing more RANSAC iterations.

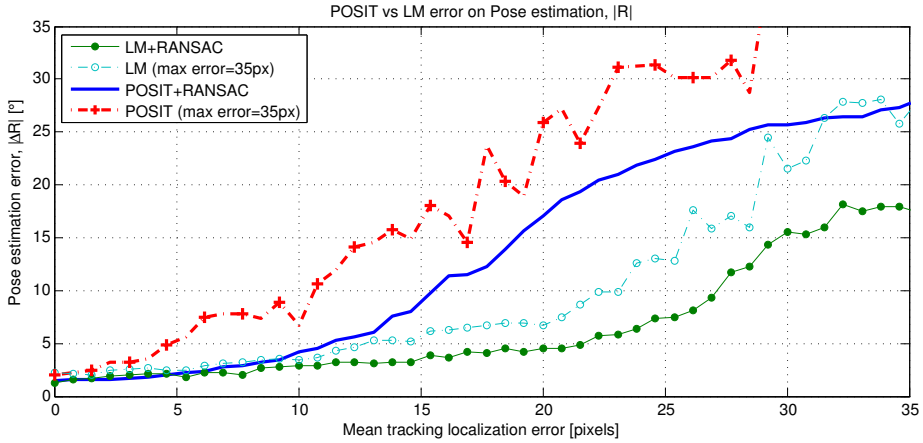


Figure 5.11: Comparison of pose estimation error using POSIT or LM, with and without RANSAC. The error threshold is set to 35 pixels. For the curve without RANSAC, the max feature error is also set to 35 pixels

5.4 Model extension and correction

The 3D model was initially created using a single pair of stereo images of a frontal face. This model is incomplete and the points may contain noise. During the execution of the algorithm, this model is extended and corrected adding new information that can be extracted from successive pairs of stereo frames. As head rotates it presents different points of view to the camera, which allows for acquiring much information from the face than that obtained from the initial pair of images.

5.4.1 Model extension with new feature points

The 3D model is initially created using a frontal view of the face. Consequently, for yaw rotations wider than $\pm 40^0$ approx, most of the points of the model are occluded. This makes necessary to augment the model with new points from parts of the face which

were initially occluded. When the number of visible points from a camera falls below a threshold, usually 10, new points are searched for. The same technique explained in section 4.2 is used to detect and obtain the 3D coordinates of the new points to be added. The search area for candidates is predefined. Many point candidates may lay outside the head itself. Those points are filtered attending to face shape and size constraints mentioned above.

The 3D coordinates of the new points to be added are referenced to the camera coordinate system. We first convert their coordinates to the model reference system, which is now defined as its estimated pose, $\mathbf{P}_t$. This pose $\mathbf{P}_t$ includes an estimation error, which thus is included in the position of new points inside the model.

5.4.2 Model correction based on bundle adjustment

The 3D points taken during model creation and added later are subject to error derived from stereo correspondences. In addition, the newly added points to the model also inherit the error of the pose estimation at the frame of addition. In order to get a better fitting of the model to the face, a bundle adjustment (BA) optimisation [Triggs 99] is used to refine the 3D model. This corrects the 3D point coordinates of the model and the poses at which any point has been added. To save on computational load, this stage is only applied at certain *keyframes*, t_k when a minimum movement has been detected in the pose, and only during certain time after model creation. The process is also executed after points have been added to the model. Each keyframe's pair of images are saved, along with the 2D projection $\mathbf{x}^r_{i,t_k}, \mathbf{x}^l_{i,t_k}$ of the model points and the estimated pose $\mathbf{P}_t$.

The BA process refines the values of the 3D model points $\mathbf{X}_i$, the past pose estimations $\mathbf{P}_i$, $i = 1, 2, \ldots$, as well as the 2D projections of model points on the images. The input to BA at each keyframe t_k is

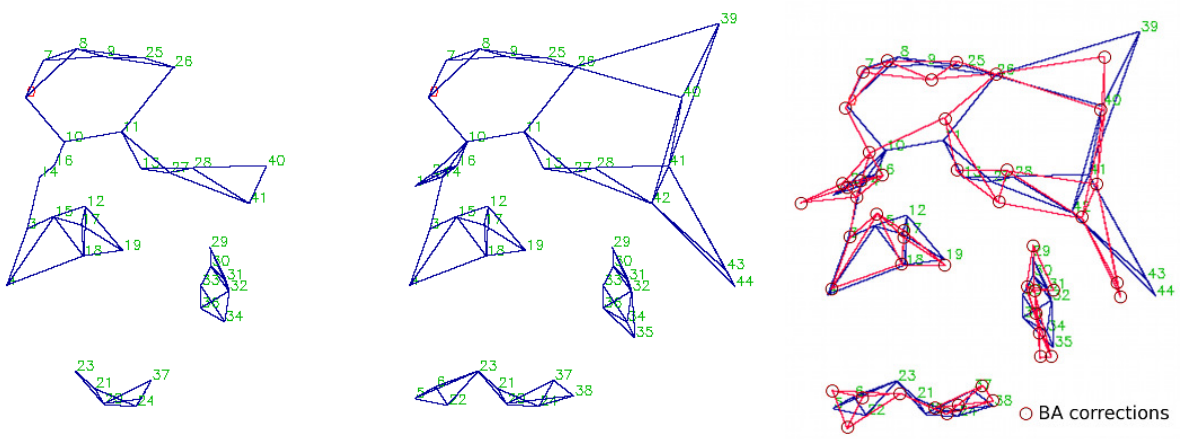
$$\begin{array}{rl} \mathcal{P}_k = & [\mathbf{P}_{t_0}, \ldots, \mathbf{P}_{t_k}] \\ \mathcal{X}_k = & [\mathbf{X}_{t_0}, \ldots, \mathbf{X}_{t_k}] \\ \vec{x}_k = & [\mathbf{x}^{\{r,l\}}_{i,t_0}, \ldots, \mathbf{x}^{\{r,l\}}_{i,t_k}],\ i = 1, \ldots, N. \end{array} \tag{5.20}$$

The error function to minimise in BA is

$$\begin{array}{c} \epsilon = \vec{x}_k - \widehat{\vec{x}}_k, \text{ with} \\ \widehat{\vec{x}}_k = [\widehat{\mathbf{x}^{\{r,l\}}_{i,t_0}}, \ldots, \widehat{\mathbf{x}^{\{r,l\}}_{i,t_k}}],\ i = 1, \ldots, N. \end{array} \tag{5.21}$$

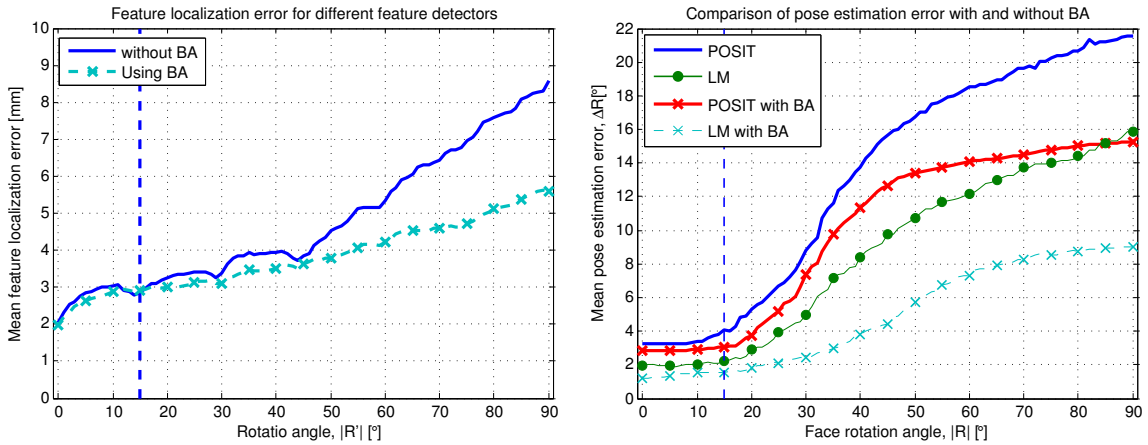
where $\mathbf{x}^{\{r,l\}}_{i,t_j}$ is the measured 2D projection of point $\mathbf{X}_i$ at keyframe t_j, and $\widehat{\mathbf{x}^{\{r,l\}}_{i,t_j}}$ is the prediction of its position after re-projection of $\mathbf{X}_{i,t_j}$ over the camera images $I^{\{r,l\}}$. The process extends until the re-projection error ϵ falls below a desired threshold.

After the model is increased on both sides and no more attempts to add new points are needed, the bundle adjustment is stopped. At this point, the residual error in the 3D model estimates is small enough that we can use the corrected model for accurate pose estimation. Figure 5.12 shows the initial model, the extended, and the corrections carried out to the model by the bundle adjustment. It can be noticed how corrections are specially needed for the new added points on the laterals of the face. The pose estimation improvement can be observed in figure 5.13. It is especially noticeable for rotations over 30°, when new points have already been added to the model.



(a) Initial model (b) Extended model (c) Corrected model

Figure 5.12: Initial 3D model, extended one, and bundle adjustment optimisation. The red lines shows the corrections done by bundle adjustment. It can be observed how the lateral points suffer bigger corrections. Points 39, 43 and 44 on the right lateral has important corrections



(a) Feature localisation error (b) Pose estimation error

Figure 5.13: Comparison of pose estimation error using BA and without, both for POSIT and LM algorithms

5.5 Conclusions

In this chapter, a novel technique of re-registering was proposed, based on the view-point of a feature from both cameras. It avoids the typical drifting problem by anticipating its appearance by using textures previously captured with one camera and tracking with the other. This technique allows to maintain feature localisation accuracy on the full yaw rotation range.

For rotations in pitch and roll a template warping simulates the patch appearance variation under 3D rotations on a 2D image.

Two techniques have been tested for pose estimation. POSIT is a very fast method, although LM demonstrated much better accuracy. This makes LM algorithm preferable for offline processing, while POSIT can be used if strict real time requirements exist.

Finally, the 3D coarse face model created at initialisation is extended with new points,

adding new parts of the face that were not visible at model creation. To refine the initial model and reduce the error that the points added later may have, a bundle adjustment correction process is executed. BA is especially important to correct the pose with wide rotations. It is also noticeable how the BA can improve pose estimation when a non-robust estimator such as POSIT is being used.

Chapter 6

3D Gaze Estimation

Chapters 4 and 5 presented methods to estimate the pose of the driver's face. However, the information on where the face is pointing does not take into account the user's eyes, which indeed have a very important role in the individual's perception and attention to the visual world. In chapter 2 it was explained that a comprehensive study of the user's gaze fixations necessarily needs taking into account both the face pose and the eye direction. This chapter describes a method to obtain this eye direction and the composition of it with the face pose to obtain the final gaze pose estimation.

Gaze is defined as the line of sight of a person and represents the person's focus of attention. For an average rested individual, the movement of the eyes is much faster than that possible by the head [Henderson 03]. Moreover, in a general human behaviour, when the person already knows the location of an area of interest well, focusing of gaze on that area is mainly done by moving eyes as far as the field of view allows for it. On the other hand, if the area is somehow new to the user or implies a high movement of the eyes, the fixation is mostly accomplished with a head movement, so that the eye direction is closer to the face direction, providing a wider range of eye movement around the area being explored [Henderson 98]. This means that while driving many of the undertaken actions to focus attention on different well-known areas of the vehicle or the environment are performed by eye movement, trying to leave the face as static as possible, pointing to the front. This is the case of the rear mirrors and IVIS such as GPS, tachometer or hands-free. Looking at the road, outside the cabin, where the environment is much unpredictable, involves much head movement. Looking at the rear mirror on the laterals of the cabin also forces the user to move the head because they are outside range for the eyes. However, since they are at very well known locations, many drivers turn the head as little as possible, moving eyes as much as they can feel comfortable. These behaviours also depend on the experience of the driver [Mourant 72].

All this makes eye direction a very important component of the gaze, because it is not possible to infer a relation between face pose and eye direction. [Henderson 98] cited a few important facts to understand eye movements in scene viewing. At least two of them are relevant to the case of study. First, eye movements are critical for efficient and quick acquisition of visual information during complex visual tasks. Second, eye movement data provide an unobtrusive, online measure of visual and cognitive information processing.

Analysing gaze data and the eye scan path over the scene, it is possible to determine which regions were looked at. However, we cannot be fully confident that these specific regions were fully perceived. To date, there is not a simple way of knowing what the brain is doing during a particular visual scan of the scene. Ideally, we would have to record not only the point of user's gaze, but also user's brain activity [Duchowski 02]. This means

that, even thought gaze is very accurately estimated, we can not be sure of what part of the scene within a narrow field of view is really the area of interest. It is possible to visually fixate on one location while simultaneously diverting attention to another. To establish an accurate point of fixation within less than 1° of accuracy, it would be necessary to study not only the eye direction, but also the saccadic eye movements which ultimately define the focusing region of interest [Hoffman 95]. Studying saccadic movements is a very complex task which is far behind the area of interest in this thesis.

The reasoning of developing a gaze estimation on top of the face estimation in this thesis is to provide a correction algorithm respecting the face in order to be able to identify where, within a set of *a priori* defined landmarks locations, the driver is looking at. The psychologists who command the simulator define these landmarks or areas of interest from the point of view of the driver's behaviour study. Typically, these refer to the road (road itself, overtaking vehicles, intersections), off-road locations (traffic signals and lights), lateral rear mirrors and IVIS (GPS, hands-free, tachograph, on-board computer, radio, etc). Thus, the accuracy of the gaze estimation algorithm must allow to distinguish which of these landmarks is the driver's point of interest or fixation area.

Gaze fixation is not considered a main contribution in this thesis. The objective of this chapter is not to create an accurate gaze estimation system, but to provide a mean to obtain a coarse gaze estimation in order to accomplish the objective of providing driver distraction behaviour statistics. The rest of this chapter explores the developed algorithm to obtain the coarse eye direction, and how to infer the landmark of attention joining this information with the face pose estimation. Simulator test scenarios, exercises defined by the team of psychologists and results are presented in the next chapter.

6.1 Eye direction estimation

To obtain the fixation areas, it is necessary to take into account both the face pose and the eye directions. We calculate the eye direction, $\vec{e}$, with respect to the model coordinate system. Consequently, the gaze $\mathbf{G}$ is computed as the composition of the both measurements. This composition gives a 3D gaze, which can be defined as an unitary 3D vector in the scene and an origin point. This origin lays within the face, and is typically defined as the centre point of both eyes. In our case, we compute the gaze origin as the 3D face model central point m_o plus a known offset to the centre of the eyes, $\vec{e}_{off}$, which is calculated at initialisation. At the gaze estimation step, face pose is already known because it has been calculated during the previous stage, and m_o is given by the face pose translation vector. Figure 6.1 shows the difference between gaze and face pose, and figure 6.2 depicts the face with different fixation points with the same pose.

The steps of the coarse algorithm for gaze estimation are shown in figure 6.3. It consists of the following steps:

* **Initial 3D face model creation**: 3D face model is first created, as explained in chapter 4.

1. **Initial eye detection**: At the model creation stage, some characteristic points around eyes are detected within the face on both images.

* **3D face pose estimation**: In a loop, face pose is estimated from frame to frame as described in chapter 5.

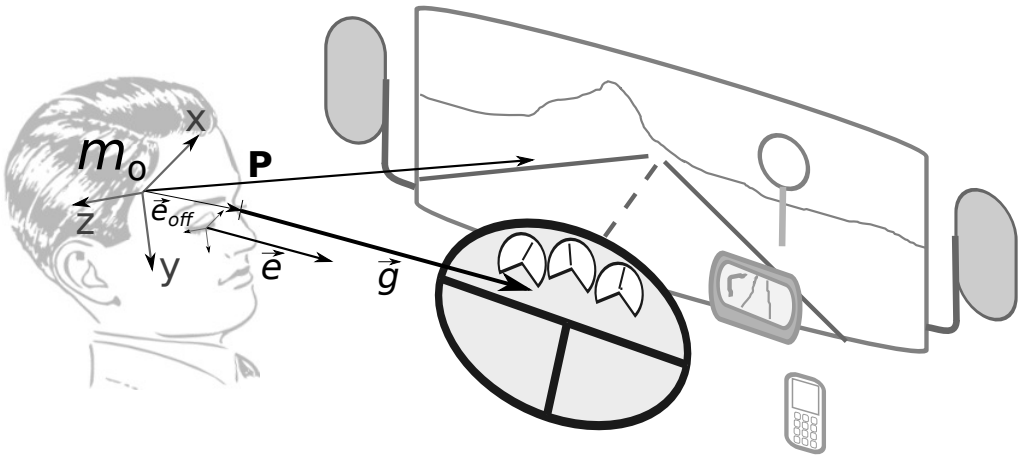
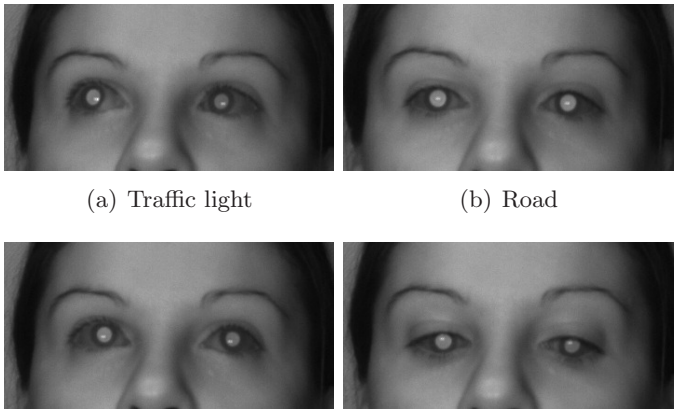


Figure 6.1: Difference between gaze and face pose



(a) Traffic light (b) Road

(c) Overhead signal (d) On-board computer

Figure 6.2: Gaze with fixation at different locations

2. **Eyes 2D tracking**: The 3D coordinates of the characteristic points are added to the 3D face model. These points are not used for face tracking and pose estimation, but for pupil localisation only.

3. **Pupil tracking**: At each frame, after the face pose estimation stage, a new stage calculates the eye direction, and composes gaze. First, the pupils, and rest of characteristics around the eyes are tracked from frame to frame, in a similar way 3D face features are tracked.

4. **Pupil centre localisation**: Pupils exact centre position is located for each eye using the integral projections algorithm and a Gaussian approximation.

5. **Pupil displacement calculation**: The eye direction is calculated as the relative displacement of the pupils centre with respect to the 3D coordinates of themselves and the rest of the characteristic points around the eyes.

6. **Gaze estimation**: Gaze is computed rectifying the face pose estimation with the eye direction estimation.

7. **Fixation classification**: The fixation point is classified based on the gaze estimation, to infer to which of a set of interest areas the subject is looking at.

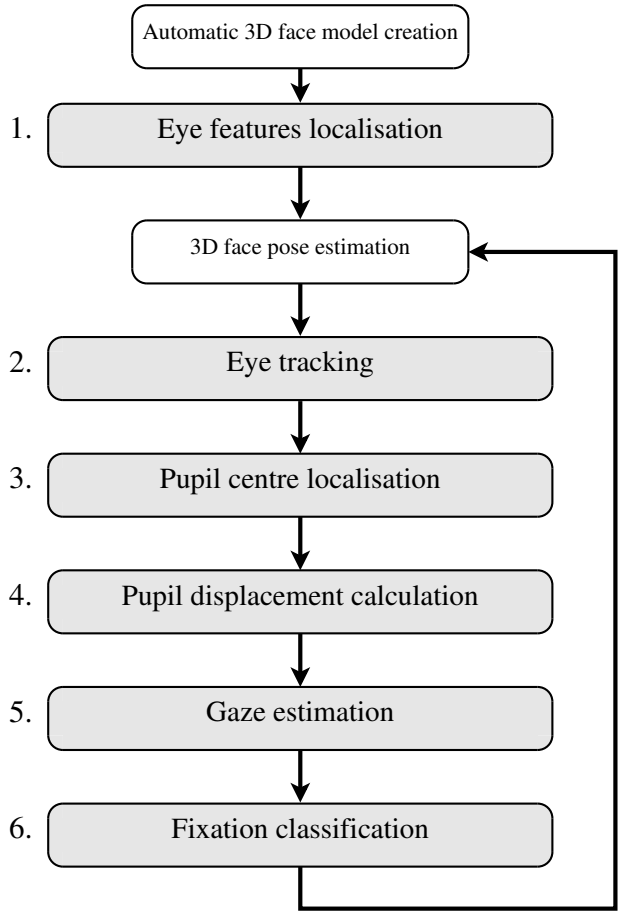


Figure 6.3: Main blocks of the gaze estimation algorithm.

These steps are explained in the following sections.

6.1.1 Initial eye features location

The eye direction is computed comparing the pupil position on each frame with the original position it had when the model was initialised. After the 3D face model has been created, a set of predefined eye features corresponding to characteristic points around the eye, typically eye corners and pupil, are located on both frames using *Stacked Trimmed Active Shape Models* (STASM) [Milborrow 08]. The 3D coordinates of these features are stereo reconstructed based on this information, and stored along with the 3D model information. Three features are obtained, as shown in figure 6.4.

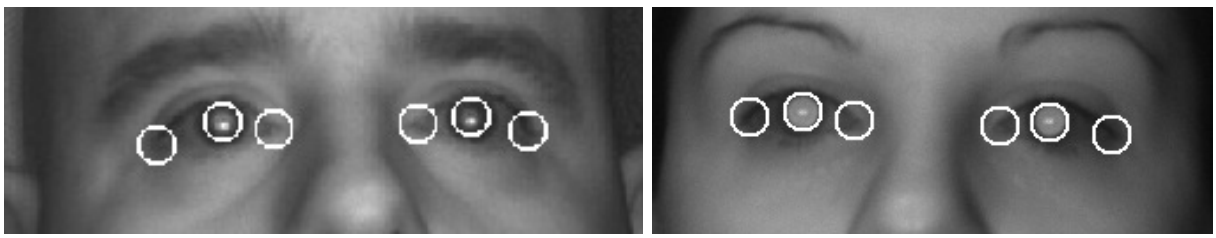


Figure 6.4: Eye features positions obtained using the STASM algorithm

6.1.2 Eye tracking

Frame to frame, eye features are located within the face in the same way the rest of the model features are searched for, as explained in section 4.1.2. The only different is that the pupil patches identified during the initial eye features location step are smaller, so that the patch only slight bigger than the pupil. Specifically, only the two smaller patches of the three sizes following the multisize matching scheme (see section 4.1.2) are used.

If the tracking fails for any eye then it is considered either closed of occluded, and consequently the rest of steps are not accomplished. If the failure occurs for only one eye, the measure of the other is considered. In case both eyes fail, the previous measurement is applied as eye direction. The tracking can fail for two reasons: the pupil feature correlation result is too low, typically below 50%, or the localisation point is too far from the model projection location after pose estimation, that is, the reprojection error is high, and the feature is considered an outlier.

6.1.3 Pupil centre localisation

After all eye features are located, pupil is located so it can be compared with the initial position stored in the 3D model.

To obtain the exact pupils centre location, we apply the following steps to the image patches around each eye, P_e^r and P_e^l, for the right and left eyes respectively.

Eye image preprocessing

The objective of this stage is to improve the image quality needed to extract the exact pupils centre position. Image eye is filtered by a hat transform [Jalba 04]. This transform subtracts from the original image an image to which a closing operation is applied. The hat transform erases most of the bright little parts inside the eye and smooths the images. To improve robustness against illumination variations, an uniform equalisation is then used. Eye features can be analysed more easily in the equalised image than in the original one. Figure 6.5 depicts three different images: The first one is the original image detected by patch correlation with the texture stored in the model, the second one is the result of the hat transformation, and the third one is the equalised image.

(a) Original eye detection (b) Hat transform (c) Equalised image

Figure 6.5: Eye images preprocessing steps for gaze estimation

Integral projections and Gaussian model

We define the integral projection of an image as the average of the pixels along parallel straight lines on the image, disposed in a particular direction. Given an eye patch P_e with dimensions $W \times H$, the horizontal integral projection $\mathcal{P}_H(v)$ is the average of each row

v, and the vertical integral projection $\mathcal{P}_V(u)$ is the average of each column u. These are defined as

$$\mathcal{P}_H(v) = \frac{1}{W} \cdot \sum_{u=0}^{W-1} P_e(u,v); \qquad \forall v = 0, \ldots, H-1, \tag{6.1}$$

$$\mathcal{P}_V(u) = \frac{1}{H} \cdot \sum_{v=0}^{H-1} P_e(u,v); \qquad \forall u = 0, \ldots, W-1. \tag{6.2}$$

The eye patches have a circular intensity distribution due to the pupil characteristics. This makes the projective integrals in both axes have a Gaussian distribution with different variance, median and mean, depending on the pupil size and eye openness. For this reason we use a Gaussian function to model the pupil. The Gaussian function to model $\mathcal{P}_V(u)$ is defined as

$$f_V(u) = \frac{1}{\sqrt{2\pi\sigma_V^2}} e^{\frac{-(u-\mu_V)^2}{2\sigma_V^2}}, \tag{6.3}$$

where μ_V and σ_V are two parameters that determine the open eye shape, and are determined each frame. The Gaussian mean μ_V indicates the vertical position of the centre of the pupil, and σ_V gives a measurement of its vertical size. The mean μ_V and variance σ_V are calculated as

$$\mu_V = \sum u\mathcal{P}_V(u), \tag{6.4}$$

$$\sigma_V = \sum (u-\mu)\mathcal{P}_V(u). \tag{6.5}$$

The Gaussian function $f_H(v)$ that models $\mathcal{P}_H(v)$ and its μ_H and σ_H are defined in an equivalent way.

The pupil positions are given by the maximums of the Gaussian approximation to the integral projections on the u and v directions.

Figure 6.6 shows the results of applying integral projections to an eye patch.

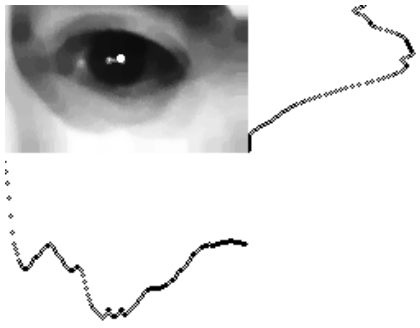


Figure 6.6: Pupil centre localisation using integral projections

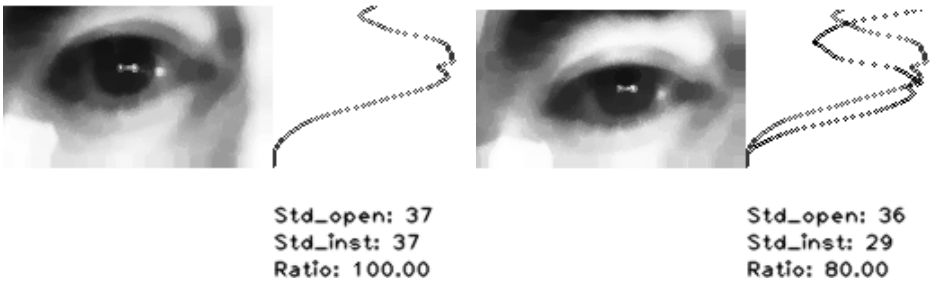
Evaluation of pupil opening

People blink and close their eyes very frequently, and specially when changing their fixation point from one location to another. These blinks must be detected to avoid the eye direction error while the eye is closed or partially open.

The eye opening or the pupil height is evaluated measuring the standard deviation of the Gaussian that models the $\mathcal{P}_H(v)$. Figure 6.7(a) shows a wide open eye. To evaluate the

eye opening percentage, the initial opening is calculated at model initialisation. With this value it is possible to evaluate the instantaneous percentage of eye opening from the ratio between the pupil's height in the current frame and the one obtained in the initialisation. Figure 6.7(b) shows another example, where the eye is open to a lesser extent.

When the eye is detected to be 50% closed or more, the eye direction is not determined, and we use the last measurement before closing. When driver's head turns and the algorithm does not detect any pupil the gaze is set to point to the rear mirrors, depending on the face pose estimation.



(a) (b)

Figure 6.7: Sample of aspect-ratio of the eye opening, showing two different eye opening

6.1.4 Pupil displacement

The pupil horizontal and vertical displacement on the image, d_H and d_V, are calculated as

$$d_V = \mu_V - u_{e,0}, \tag{6.6}$$

$$d_H = \mu_H - v_{e,0}, \tag{6.7}$$

where $(u_{e,0}, v_{e,0}) = \mathbf{x}_{e,0}$ is the initial pupil centre on the image stored in the model. However, d_H and d_V do not only depend on the pupil position, but also on the pose $\mathbf{P}_t$ at the frame being evaluated because the point of view of the eye changes with pose, and consequently the projections distances d_H and d_V. Since the 3D face pose is known from previous algorithm stages, it is possible to rectified d_H and d_V to obtain the displacements across pose, d_{Hp} and d_{Vp}, which suppose the face is at the same frontal position than at the initialisation. The displacements across pose are computed as

$$x_{ze} = (R_t \mathbf{X}_e^{(\mathcal{M})} + T_t) \cdot \vec{v}_z, \tag{6.8}$$

$$d_{Hp} = \frac{d_H x_{ze}}{f_c \cos \alpha_y}, \tag{6.9}$$

$$d_{Vp} = \frac{d_V x_{ze}}{f_c \cos \alpha_p}, \tag{6.10}$$

where $\mathbf{X}_e^{(\mathcal{M})}$ is the initial eye 3D position in the model, $\{R_t, T_t\}$ is the pose at the current frame, $\vec{v}_z$ is the unitary vector in the perpendicular direction to the camera, α_y and α_p are and the yaw and pitch rations of the face, $\cdot$ denotes the product vector, and x_{ze} is the distance to the camera if the pupil where at its original model position. These equations give an approximation of the pupil 3D position relative to its initial 3D position stored in the model, as shown in figure 6.8.

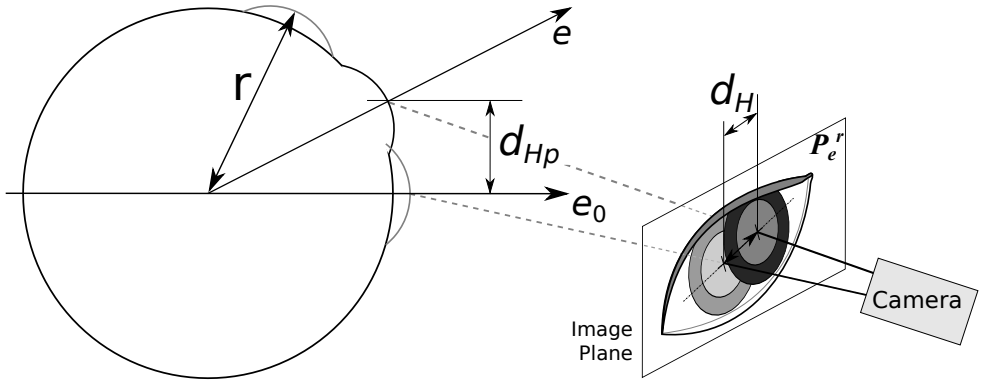


Figure 6.8: Transformation from d_H to d_{Hp}. The radius r of the eye is user dependent

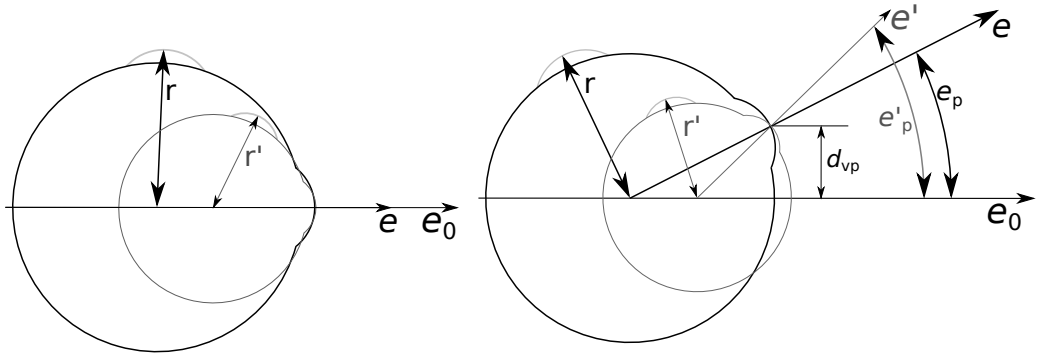
6.1.5 3D Gaze vector

Eye direction is defined as a yaw and pitch offset angles over the face pose, e_y and e_p. The offsets are function of the horizontal, d_{Hp}, and vertical, d_{Vp}, deviation across pose of the pupil centre from its original model position:

$$e_y = f(d_{Hp}, r) = \sin^{-1}(\frac{d_{Hp}}{r}), \tag{6.11}$$

$$e_p = f(d_{Vp}, r) = \sin^{-1}(\frac{d_{Vp}}{r}), \tag{6.12}$$

where r is the radius of the ocular globe, and it is different for each user, and unknown. It must be determined experimentally. Figure 6.9 depicts two different eye directions vectors $\vec{e}$ and $\vec{e}'$ for users with different ocular radios, r or r'.



(a) Eyes looking front, $d_V = 0$ (b) Eyes with a pupil displacement d_V

Figure 6.9: Effect of the different ocular radio r to the eye direction estimation

To calculate the ocular radius r, a calibration process is done during the first minutes of operation, based on *saliency* fixation areas locations. During the first minutes of execution, the driver will look to both rear mirrors, the road and the on-board computer, as he is forced to do so by the driving exercises, defined in section 7.5.1.

The position of these elements is well known in the 3D space around the cabin. The moments in which the user is looking at these points are given by the face pose estimation and pupil variation from central position. When the user is looking at any of these points, we obtain the values d_{Hp} and d_{Vp}. Since the 3D fixation positions and the face

pose are also known, we can also calculate e_y and e_p. For each measurement of the parameters $(d_{Hp}, d_{Vp}, e_y, e_p)$, an estimation r_i is obtained. With these values we compute the parameter r using recursive mean squares. The calibration time extends until the RMS error of r_i drops below a threshold. It usually takes around 2 minutes, although it depends on how often the driver looks at the mirrors and at the computer.

The gaze $\mathbf{G}$ can be specified as a parametric line by its point of origin, $T_g = (x_t, y_t, z_t)$, an unitary vector, $\vec{g} = (v_x, v_y, v_z)$, and a free parameter s. Recall that $\vec{e}_{off}$ defines the gaze origin within the model reference system, and that it is subject constant and calculated at model initialisation. Then

$$T_g = T + R \cdot \vec{e}_{off}, \tag{6.13}$$

$$\vec{g} = -RR_x(e_y)R_y(e_p)|V_0|, \tag{6.14}$$

$$\mathbf{G} = T_g + s\vec{g}, \tag{6.15}$$

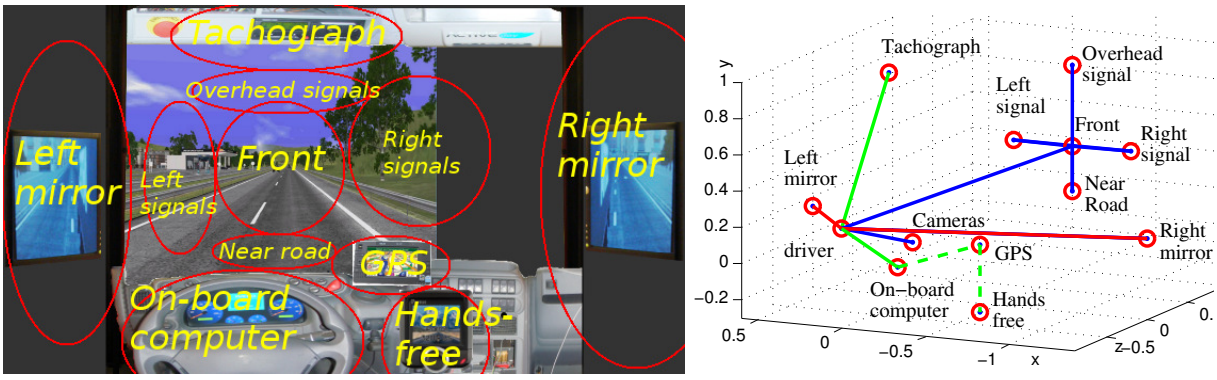
where $\{R, T\}$ is the face pose at the current frame, $R_x()$ and $R_y()$ are the rotation matrices around the x and y axis, and $\vec{V}_o$ is the initial face offset vector, defined in section 4.2.2.

6.1.6 Gaze fixation and classification

The objective of this chapter is to determine the fixation areas of the driver, to know where of a set of key areas is she/he looking at. The possible key fixation areas are shown on figure 6.10, and are:

- *Front*: the road itself and traffic ahead. Victor and Joanne, based on experiments on various simulators and on real traffic, defined this as an area between 16° and 20° in diameter centred on the road [Victor 05].
- *Left and right signals*: denote the signalling on sides of the road, overtaking cars, crosses or other objects present in the proximity of the truck. When the driver is looking at any of these points, the fixation is slightly diverted horizontally to the left or to the right.
- *Lateral rear mirror*: the external rear mirrors located at both sizes of the cabin. Many times it is not possible to localise the pupils when the driver is looking there, but it is easily recognisable when the driver looks at these points because he/she needs to turn largely the head horizontally.
- *On-board computer, GPS and Hands-free*: These are the on-board IVIS. Usually the driver tries to look at them with very little head movement, to not lose attention to the road.
- *Tachograph*: This IVIS is located overhead, over the windscreen, and looking at it requires head movement.
- *Overhead signalling and near road*: Looking at these points requires no head movement and very little vertical pupil displacement, so it is difficult to distinguish when the driver is looking there from the front road itself.

The cameras position inside the cabin is fixed, and the geometric layout of the simulation room, cabin and projection panels are known, so the 3D centroid $\mathbf{Y}_k$ in the scene of each of the regions described above, can be measured and referenced to the right camera frame system. The fixation area is calculated as the closest key fixation centroid



(a) 2D view of the key fixation areas

(b) 3D positions of the centroids of the key fixation areas

Figure 6.10: View of the set of key fixation areas

to the gaze line. Following equation (6.13), the minimum Euclidean distance between a 3D centroid $\mathbf{Y}_k = (x_k, y_k, z_k)$ and the line defined by the gaze can be expressed as

$$d_k = \frac{|\vec{g} \times (T_g - \mathbf{Y}_k)|}{|\vec{g}|}, \tag{6.16}$$

where $\times$ denotes the vector product. Once the minimum distances to each key fixation area are computed, a classification is performed based on this parameters and area sizes.

6.2 Conclusions

In this chapter we have presented a simple method to calculate a coarse eye direction based on the pupil positions compared to their initial positions at the model creation. This eye direction can be added to the face pose to generate a gaze estimation.

A calibration process must be carried out during the first minutes of execution in order to calculate the ocular radius, which is user specific. This calibration is transparent to the user, thanks to some scheduled movements that the driver is required to do during the driving exercises.

Extended testing methodology and results for the gaze estimation are presented in section 7.5.

Chapter 7

Tests and Results of the Driver Distraction Monitoring Application

This chapter presents the results of the algorithm proposed in this thesis, applied to the available database of videos taken from a naturalistic truck simulator, under the facilities of the CABINTEC project [CABINTEC 11]. It is divided in two main parts. First, we carry out an analysis of our face pose estimation proposal. This analysis focuses on some of the parameters that can be adjusted in the algorithm, such as the feature patch size and thresholds. It also compares the error for the different techniques seen in sections 4.1, 5.1 and 5.3. As one of the requirements is real-time performance, processing times for the different configurations are also examined. Finally, the addition of gaze to the pose leads to the analysis of distractions in the video sequence, where the drivers' behaviour is studied.

7.1 Hardware and software description

The test environment is a naturalistic truck simulator, as shown in figure 7.1, which very accurately recreates day and night time driving conditions. The simulator itself is a real truck cabin, motorised with actuators to simulate driving motion. Three wide projectors outside the cabin show the scene. The two lateral rear mirrors are also screened, so the driver can look at them to check the traffic behind.

The stereo cameras have a base line of 20 cm, and are located over the dashboard behind the driving wheel, at a distance of between 60 to 100 cm to driver's head. The view of both cameras is not parallel, but have a little convergence towards the centre, to point the driver's face, making and angle of 15° between them. The stereo rig is calibrated using the Camera Calibration Toolbox [Bouguet 10] for MATLAB®.

The capture system is formed by two synchronised Basler Scout family FireWire™ cameras and two pulsed IR illuminators synchronised with the cameras. The captured video is high resolution grey scale data at 30 frames per second. Each camera has a 9mm lens on a 2/3" sensor. Although the face size is only around 300×350 pixels under these conditions, the images size are 1392×1040 pixels, allowing for a wide range of movement inside the camera field of view. Most of the face images shown in this document have been cropped for convenience.

The algorithm has been tested in a Intel® Core™2 Quad® Processor running by Kubuntu 9.10, and equipped with a ATI Radeon™ HD 4500 Series graphic unit from AMD. All code is written in C++, and parallelised using threads. Most of the specific

vision operations have been programmed using the OpenCV library [Bradski 08]. The bundle adjustment and Levenberg-Marquardt algorithms are coded using the libraries provided by Lourakis and Argyros [Lourakis 09, Lourakis 04].

(a) truck simulator used to evaluate the algorithm under real night time driving conditions, showing camera layout.

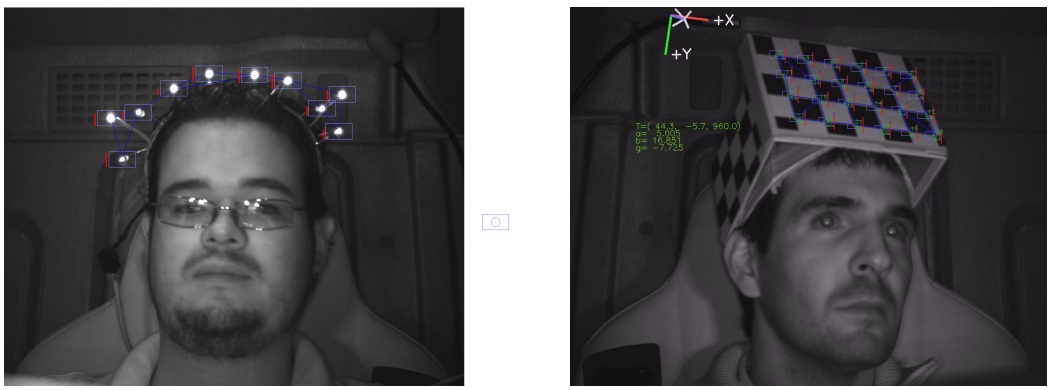
(b) View of the road, cameras and GPS from the driver's position

Figure 7.1: Track simulator used to record the video sequences

7.2 Ground-truth

The ground-truth (GT) data has been obtained for six different users, using video sequences more than ten minutes long each. The sequences were recorded within a very high immersion environment and simulating common driving disturbances, such as phone calls, handling the GPS and takeovers, which result in frequent head movements.

Two different methods have been used to generate the GT data. In some of the videos, we obtain the GT using a light pattern installed on a coronet or tiara placed on the user's head, as shown in figure 7.2(a). On other videos, a calibration pattern similar to those used for camera calibration was attached to the head, used a hat. See figure 7.2(b). The GT is calculated using MATLAB®, and its output is estimated to have an error below 0.5°. In both cases, the pattern is adjusted to the head, and treated as a disembodied rigid object.



(a) GT using a light tiara

(b) GT using a calibration pattern

Figure 7.2: Ground-truth methods

7.3 Performance evaluation

In this section we study the performance of the different methods presented in this thesis: model creation presented in chapter 4, the feature tracking algorithm described in section 5.1 and pose estimation methods, which are introduced in section 5.3. Note that the different parts of the algorithm are executed sequentially and in a loop, and would not work on their own. This means that it is difficult separate the error only caused by a single step. Where possible, the ground-truth has been used to generate the input data off a part of the algorithm, isolating its error from previous steps of execution.

7.3.1 Model creation error evaluation

Techniques for model creation are evaluated in terms of the number and the quality of the extracted features. An evaluation comparing different approaches in terms of the number of correct features and putatives has been presented in 4.1.1. Multisize Harris showed the best results. The quality of the features also refers to how well the features performs in the subsequent algorithm steps while processing such features. It can not be evaluated right after model creation, and has to be tested in terms of how those features performs during the tracking step. The testing procedure is presented in the following sections.

7.3.2 Feature tracking error evaluation

The error of tracking methods depends on several variables, of which the most important are the features being tracked. Consequently, the different approaches must be all evaluated with the same set of features.

A review of the literature on face pose estimation shows that few authors perform an evaluation of the performance of each step of their pose estimation algorithms. To evaluate the steady-state error of the tracking step separately from the entire system, [Murphy-Ch. 08] isolated the tracking error using the GT data. To find more specific methods for tracking error evaluation, it is necessary to review related works on face tracking. A quick review of published works [Cootes 01b, Cristinacce 04, Dowson 05, Nuevo 09] shows that all authors consider the performance of their face tracking algorithms as a function of the distance between the estimated position of the features and their GT position in the camera images. However, as explained in section 7.2, the GT does not provide data on local face features. In the case of the face tracking methods mentioned above, GT values are usually hand marked by a human operator.

Here, it is possible to extrapolate this data from the pose registered on the videos. The process is as follows: once the 3D model is created, we reproject the model features over the camera image using the GT pose. The reprojected positions are then used as the GT of the features localisation. Because the 3D coordinates of the features may change after the correction step (see section 5.4), the localisation error is not calculated right after the processing of a frame. Instead, the GT for the feature position is carried out by a two pass algorithm execution using the pose in the GT: first, we calculate the 3D coordinates for the features, using the techniques shown in chapter 4 and section 5.4, then we determine their projections over the image using the pose GT. This method assumes a rigid face model. Variations of the face due to gestures changes are treated as tracking errors.

Several functions have been used in the literature to evaluate the tracking error. Some authors use the *Root Mean Square* of the distance between points and their corresponding GT values. [Cootes 01b] denotes is as *RMS-PE*. Others use the mean of the point-to-point

Euclidean distance as the error measurement:

$$m_e = \frac{1}{n}\sum_{i=1}^{n} d_i, \qquad d_i = \sqrt{(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T(\mathbf{x}_i - \bar{\mathbf{x}}_i)}, \tag{7.1}$$

where $\mathbf{x}_i = (u_i, v_i)$ is the position on the image for feature i, estimated by the tracking algorithm, $\bar{\mathbf{x}}_i$ is the projection of the calculated GT coordinates for feature i, d_i is the error measurement, and n is the number of features. [Stegmann 05] used two different distances: between the points and their GT, and between the points and the curve that links the GT values. The calculation method used in this thesis is the same used by [Cristinacce 06] and [Nuevo 09]. They introduce a scaling factor in the measurements, which depends on some reference size of the object. In their work, they use the distance in pixels between the eyes of the person on the image when the face is frontal to the camera.

This scaling factor compensates for the apparent variation in size when the person is closer or further away from the camera, and for the different size of the face of different subjects. However, this distance may not be accurate when the face rotates around the vertical axis. Since we use a stereo camera in this thesis, it is possible to convert pixels to a distance in millimetres, making the error figure independent of the distance to the camera and of the face size. Equation (7.1) can be expressed in millimetres as

$$m_e = \frac{1}{n}\sum_{i=1}^{n} d_i, \qquad d_i = \sqrt{(\mathbf{X}_i^z - \bar{\mathbf{X}}_i)^T(\mathbf{X}_i^z - \bar{\mathbf{X}}_i)}, \tag{7.2}$$

where $\bar{\mathbf{X}}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$ is the calculated GT position and $\mathbf{X}_i^z = (x_i, y_i, z_i)$ is the 3D coordinate of feature i estimated from $\mathbf{x}_i$ in (7.1) applying the 3D conversion equation, assuming $z_i = \bar{z}_i$. The condition $z_i = \bar{z}_i$ must be applied since $\mathbf{x}_i$ is estimated over the camera image projection, and there is no z_i information. Since the tracking step assumes planar images at all times, it makes sense imposing the condition that the located feature is at the same z plane than the GT coordinate.

7.3.3 Pose estimation error evaluation

Most authors calculate the pose estimation error as the mean pose distance between the estimated pose and the GT, over the whole test set. In some cases, specially for systems that identify discrete poses (see chapter 2. State of the Art), the error is given as a percentage of correctly estimated poses, for some predefined thresholds. As for the tracking methods described in 2.1.7, it is possible to give a percentage of pose estimation success, calculated as the amount of time that the algorithm generates a valid pose. [Murphy-Ch. 08] quantified the invalid poses as the frames where the tracked head orientation deviates more than 30° from the true pitch, yaw, or roll. In such case, the pose is not included in error computations. On the other hand, the error is calculated as the mean difference between the estimated parameter and the GT value, for any of the six degrees of freedom of the face pose (x, y, z, *yaw*, *pitch* and *roll*). However, for a better comparison of errors, it is easier to show the pose error as a single value. We calculate the error p_e as

$$p_e = \sqrt{\alpha_p^2 + \alpha_y^2 + \alpha_r^2}, \tag{7.3}$$

where α_p is the *pitch* angle error , α_y is the *yaw* angle error and α_r is the *roll* angle error. In this formula, p_e only takes into account the rotation error. Although this is

not a full error measurement, it can be used for comparison and plotting, since the error derived from translation estimation is typically much smaller than that from rotation. All measurements in related literature and in this thesis are given in degrees for rotations, and centimetres for translations.

7.4 3D Face pose estimation results

The analysis of the system performance has been carried out using the available videos from the database with GT information. Generally, rotation gives much of the information about the system accuracy, so most of this section focuses on the frame to frame rotation changes ΔR, and in the absolute rotation with respect to the initial face position, R. To compare different rotations, it is necessary to take the translation into account, applying a pose correction first as it is shown in figure 7.3. The purpose of this transformation is to make the output error independent of the face translation as it is not going to be used in the comparisons. The idea is to translate the model under all the poses to be compared to a common reference, while maintaining the rotation variations with respect to the camera i.e., transform the pose variation to pure rotations. This way, translation can be safely discarded from comparison.

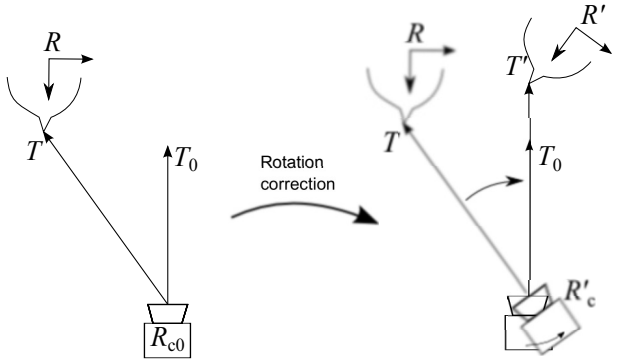


Figure 7.3: Correction from a model pose $\{R, T\}$ to $\{R', T'\}$ in order to make T' similar to the initial pose T_0

Let $\mathbf{P}_0$ be the initial pose of the face, and $\mathbf{P} = \{R, T\}$ the face pose, rotation and translation at certain frame t. If $T \neq T_0$, although R were equal to R_0 the face has an apparent rotation with respect to the camera because the point of view changes. A transformation $\{R, T\} \rightarrow \{R', T'\}$ could be applied so that the point of view of the face from the camera is the same than with translation T_0, except for the distance to the camera. Then, R and R' has the same rotation with respect to the camera. This way, R' accounts for appearance variation caused by the translation T. To express it in other way, let R_c be the right camera rotation, which is indeed the world coordinate frame system. A camera rotation $R_c \rightarrow R'_c$ could be applied so that the face is at the centre of the image, that is, $T' \approx T_0$, except for the distance. If camera distortion is discarded, the appearance of the face with camera rotation R_c or R'_c is very similar, if not equal, since this transformation does no involve a 3D projection transformation of the object. This safely discards T and only R' can be used to rotation comparison charts. Formally, $\{R', T'\}$ can be defined as

$$\begin{aligned} R' &= RR_d \\ T' &= TR_d. \end{aligned} \tag{7.4}$$

A rotation matrix R_d must be found which satisfies

$$TR_d \cdot T_0 = 0, \tag{7.5}$$

where $\cdot$ denotes the vector dot product. In practise, it is much easier to calculate R' experimentally. Before applying the 3D pose calculation algorithm, whether POSIT or LM, the set of projection points of the features are centred to the camera image. This makes the POSIT or LM to only calculate the face rotation.

In the rest of the chapter, the pose estimation parameter used is R', calculated this way, unless otherwise noted.

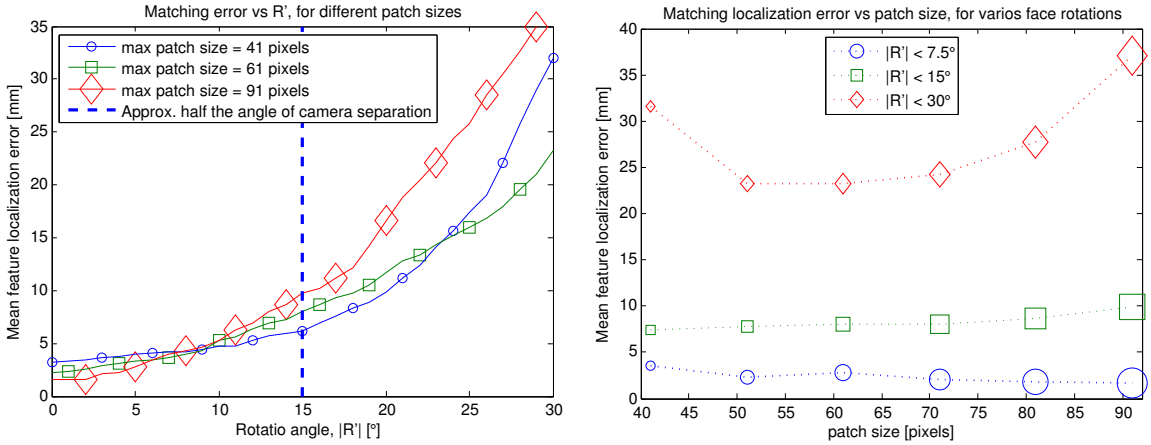
7.4.1 Performance analysis for different patch sizes

We test the performance of out system with several sets of parameters. The first parameter to be tested is the patch size. There is a compromise in selecting the patch size. Smaller patches are more specific and are less sensitive to illumination changes. However, when movements take place, smaller patches are more difficult to track, specially if motion blur appears. Blur is likely to appear if the user moves the face even slowly, due to the low ambient illumination. On the other hand, bigger patches are less error-prone since they use more information, but are also less specific to a feature characteristic texture, i.e., with sufficient size, a patch centred around the eye may include part of the eyebrow, which may not be desirable. For a bigger patch size, feature appearance changes more under viewpoint variations. The face is normally around 300×350 pixel in the image. All the videos have been batch processed with different patch sizes, from 41 to 91 pixels, in steps of 10 pixels. Those sizes represent a range of 1.5% to 10% of the face area. All patches in the model are squared. The different patch sizes have been evaluated based on the performance of the feature localisation process. For each feature, a patch on the frontal view of the face, for the given size, is stored. This patch contains the feature initial appearance, and is correlated though a search area of 101×101 pixels (10% of the face area) around the real position of the feature in the frame.

The real position is calculated using the GT pose estimation, and reprojecting the 3D model using that pose. Error is then calculated as the Euclidean distance from the reprojected points to the localised ones. All the other corrections of the algorithm (RANSAC, outlier reprojection, feature re-registering and BA) are deactivated for the tests.

The localisation error with respect to the GT for the different patch size configurations is shown in figure 7.4. Results are only shown up to 30°. During normal algorithm operation, a stored feature template is normally going to be tracked only for 15°, which is the angle between the two cameras, before feature re-registering occurs. This limit is represented by the vertical dashed line in the plot 7.4(a). As for 7.4(b), the curve for rotations $|R'| < 30°$ is unlikely to be reached by the system under normal operation. The plots show that patch size has little effect on overall tracking performance, specially for rotations under ±7.5°. They also show that below that limit the localisation error is relatively low, and thus the tracking can succeeded with no further corrections such as re-registering. The key factor to choose a patch size turns then to be the execution time. Obviously, the smaller the patch is, the faster the matching process is. Consequently, a patch size no bigger than 61 pixel is chosen, as real time performance is a requisite for this thesis.

The plots in figure 7.4 show the mean localisation error. While the error is stable and relatively low for rotations below 7.5°, the variation of the error for different features is very high. Some features perform very well, like those on figures 7.5, but others rapidly



(a) Localisation error against rotation, for different patch sizes

(b) Localisation error against patch size, for different rotations

Figure 7.4: Comparison of the matching error during the feature localisation process, for different viewpoints, $|R'|$, and patch sizes.

degrade as rotation angles increase. Figure 7.6 shows an example of this, for different configurations.

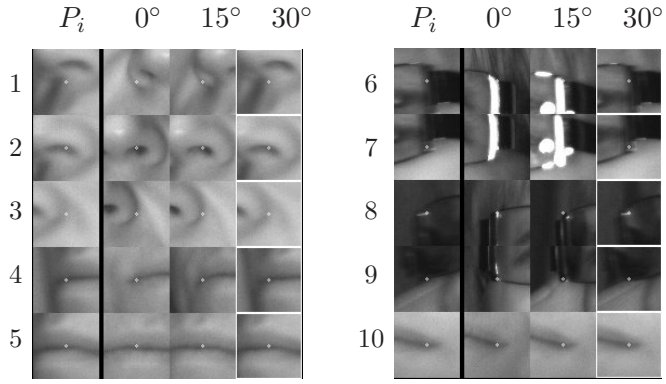
7.4.2 Performance analysis for different patch matching techniques

In section 4.1.2 we described the multisize matching method. Using multisize matching effectively reduces the feature localisation error variation, as shown in figure 7.7. We use squared patches of size $\mathbf{s}_{patch} = \{21 \times 21, 41 \times 41, 61 \times 61\}$ pixels. The correlation of the three patches are aligned and accumulated to build the resulting matching surface, $r_i^l(u, v)$, where the maximum is searched for. The correlation is undertaken over a search region of the face of size $\mathbf{s}_{search}$. However, the region of the face involved in each correlation operation itself must be bigger, as denoted by

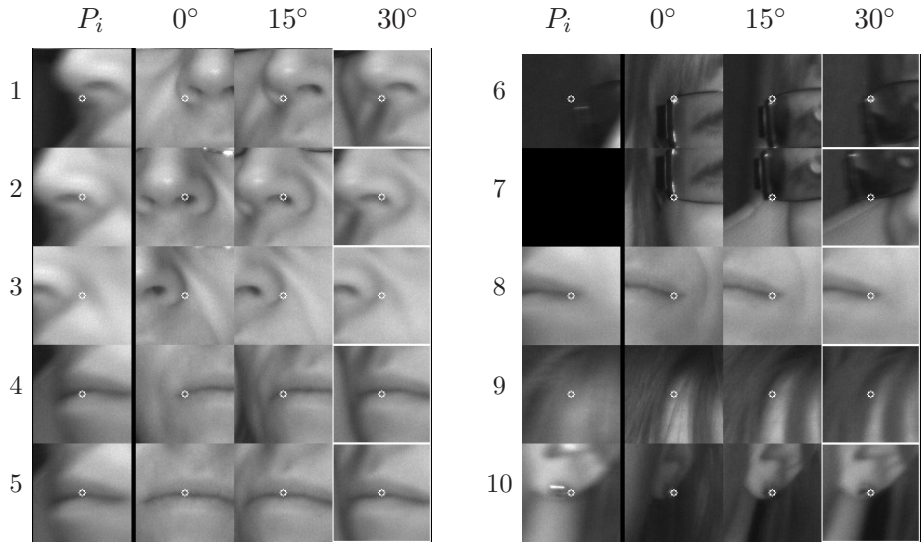
$$\mathbf{s}_{corr} = \mathbf{s}_{search} + \mathbf{s}_{patch} - 1, \tag{7.6}$$

where $\mathbf{s}_{patch}$ is the size of the feature patches (here 21, 41 or 61 pixels), and $\mathbf{s}_{corr}$ denotes the size of the area of the face involved in the correlation operation. From equation (7.6) it should be noted that the bigger the feature patch is the bigger the correlation area, $\mathbf{s}_{corr}$. This gives increased stability for the bigger patches, as more information is used. The smallest patches, on the other hand, give better performance under rotations. As a drawback, mean matching error slightly increases for rotations below 6°. This mainly happens because of the influence of the smallest patch size in the less contrasted features. A small patch contains very little texture information, while a low contrast feature also contains less information than those with better contrast. The result of the correlation of the small patch is averaged with the other patches, producing a degradation to the result. For wider rotations this effect is over passed by the sharpener correlation provided by the small patches.

Figure 7.7 also shows the reduced error variation, proving that matching with patches of different sizes is more robust for all the different features than using patches of a single size. This increased robustness makes this matching method more appropriate for feature tracking. Attending to execution times, since the smaller patches are really small



(a) $\mathbf{s}_{patch} = 61 \times 61$

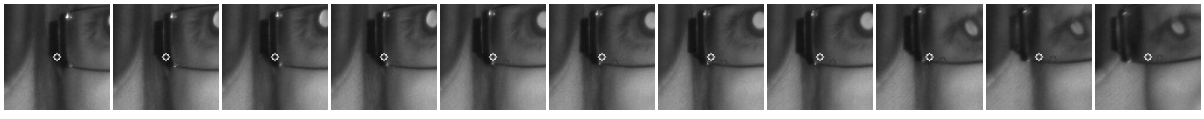


(b) $\mathbf{s}_{patch} = 91 \times 91$

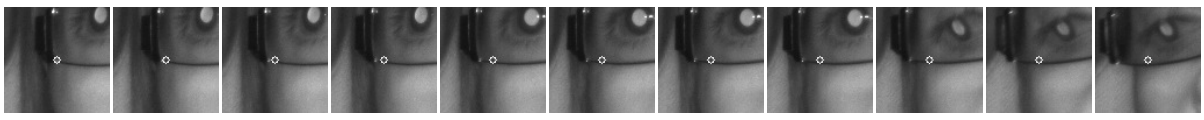
Figure 7.5: Sequence of feature views under different rotations and patch sizes. The first column shows the feature localisation results. Each row corresponds to a different feature

(a) Tracking sequence. Single-size. Patch size $\mathbf{s}_{patch} = 61 \times 61$

(b) Tracking sequence. Multisize. Patch sizes $\mathbf{s}_{patch} = \{61 \times 61, 41 \times 41, 21 \times 21\}$



(c) Tracking sequence. Single-size. Patch size $\mathbf{s}_{patch} = 91 \times 91$



(d) Tracking sequence. Multisize. Patch sizes $\mathbf{s}_{patch} = \{91 \times 91, 61 \times 61, 41 \times 41\}$

Figure 7.6: Sequence of another feature view for various small rotations, and localisation results.

— 21 and 41 pixels — the three correlations do not have a big impact in the real time performance.

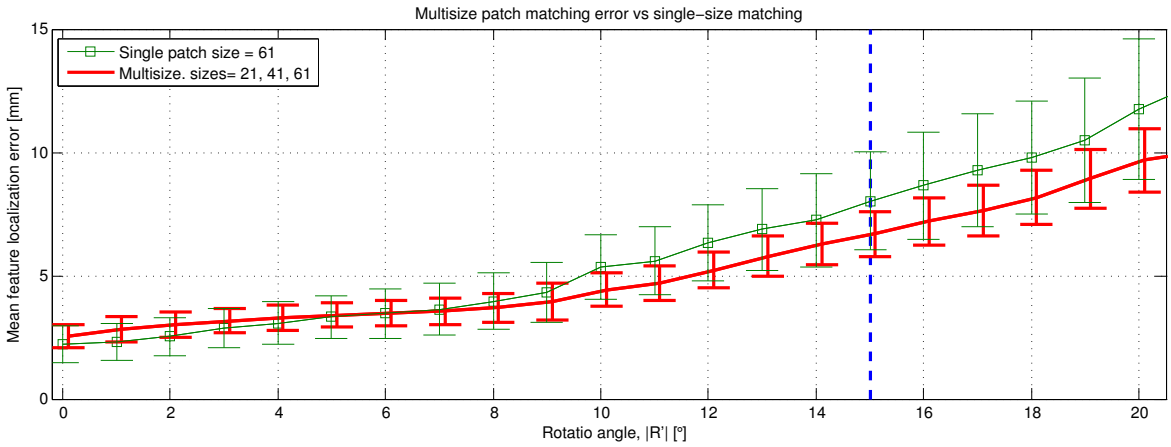
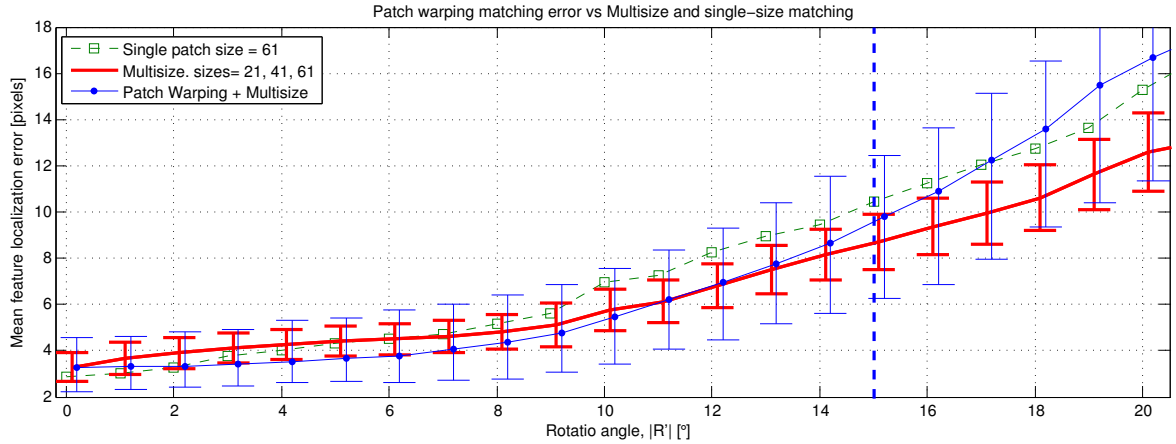


Figure 7.7: Comparison of patch correlation results using only one patch size or using the Multisize patch scheme. In the multisize matching, three sizes of squared patches are used: 61, 41 and 21 pixel

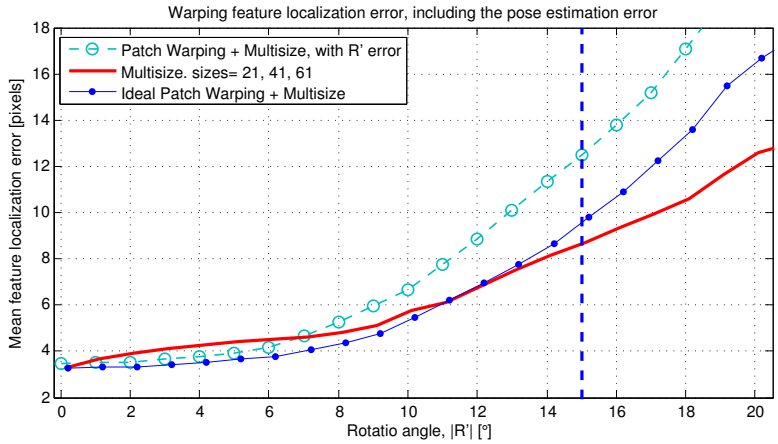
In section 5.1.1 we described the patch warping and other matching techniques. The stored feature templates are warped to the current frame rotation, according to the *pseudo-normals* extracted for each feature. Since now the intention is to obtain the error in matching introduced by warping alone, the rotation R' to apply to the warping is extracted from the GT, only for testing purposes. That way it is possible to isolate the error exclusively related to the correlation of patches, which are assumed to be correctly warped. The error would be due to the assumption of planar feature surface. The warped texture is correlated over the designated search area of the face using the multisize matching technique. Figure 7.8 depicts the localisation error using multisize matching with and without patch warping. Single size correlation error is included for reference. It can be observed how the warping technique improves matching in the interval of 2° to 10° approximately. For very small rotations it offers no improvements since there is not actually almost any noticeable transformation. For wider rotations, the assumption of a planar surface ceases to be valid, and more error is introduced by 3D deformation.

However, the real error introduced by warping is also affected by two more factors: the error in the pseudo-normals calculation, and the error in the pose estimation for the current frame, R. The former is also included in the figure 7.8(a), since the GT and the available face data do not allow for a better calculation of the pseudo-normals by any other method than manual labelling. The warping error including that added by pose estimation is compared in figure 7.8(b). The localisation error increases noticeably, specially for the wider rotations, where the pose estimation error might be bigger. For small rotations, the warping still performs slightly better than multisize matching without warping, however, the difference is negligible. Moreover, as depicted in figure 7.8(c), warping drastically increases execution time and decreases robustness. Consequently, we do not use warping for feature tracking if there is only rotation in the yaw direction.

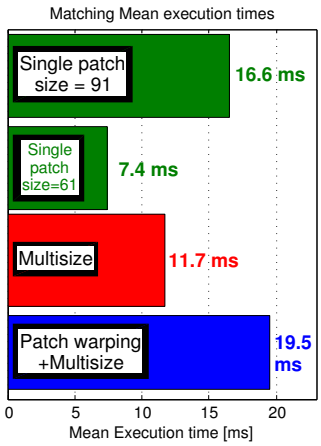
Figure 7.8(a) also shows that the error variance for warping is higher, which indicates that while some features do really increase reliability if warping is used, others perform worst. This is due to the errors in the calculation of the pseudo-normals and the different between nearly planar features and non-planar ones.



(a) Ideal case, warping R' calculated from GT



(b) Pose error en R' included



(c) Execution times

Figure 7.8: Comparison of matching results using patch warping.

7.4.3 Performance analysis for different feature detectors

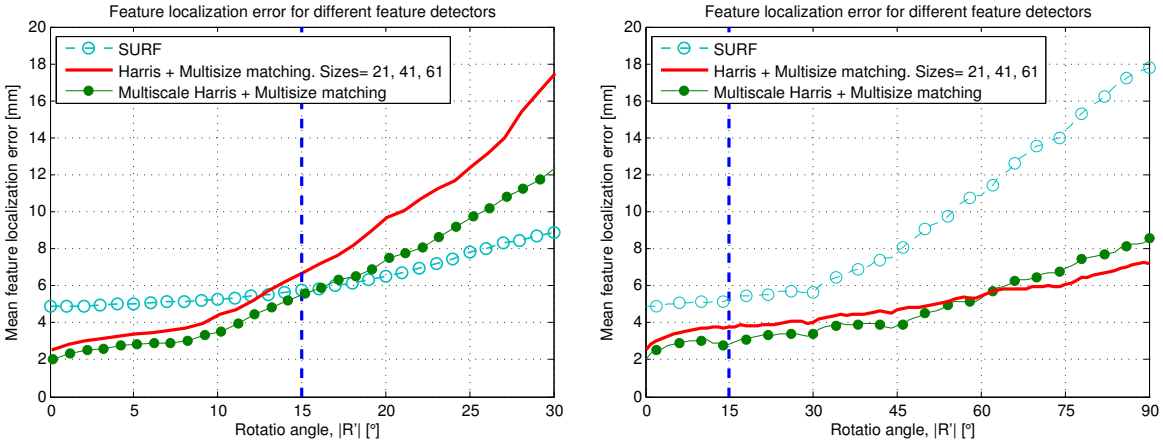
We now turn to study the incidence of the feature detectors on the tracking performance of the algorithm. Features may be detected and tracked using a combination of the methods described in section 4.1. We test three combinations. The first uses SURF for feature detection and matching. The second uses Harris and multisize matching, while the third uses a multiscale Harris detector and multisize matching.

Figure 7.9 shows that SURF does not produce good results, and is clearly outperformed by the other two combinations. This is due to the low illumination of the images and the little edges present on the face. Even so, SURF provides increased stability to face rotations. Its descriptors are more robust to rotations than the correlation templates. Figure 7.9(a) shows the result with re-registering corrections deactivated, and localisation compared to GT. Although the overall SURF error for a slightly rotated face is higher, it can be observed how this error grows more slowly than those of template correlation as the face rotation increases, to the point that for wide rotations SURF outperforms the other two techniques. However, when re-registering is enabled the template correlation techniques improves noticeably, as it is shown in figure 7.9(b). The curves indicates how the error slightly decreases at rotations where the re-registering occurs (for the stereo rig used, approximately at $\pm 15°, \pm 30°, \ldots, \pm 90°$). Under this operational conditions,

correlation clearly outperforms SURF. In part, the low performance of SURF is related to the low illumination and the repeatability of the detector under wide rotations. The re-registering process imposes that a feature is used for tracking along the whole rotation range. However, the feature projection might not be a good interest point under certain rotation. From that moment on, SURF does not extract the feature as a interest points, and fails to track it. This effect is more likely to happen if the repeatability rate is low.

Another important effect that should be noted is that, despite the re-registering, feature localisation error is a monotonically increasing function. This is due to the accumulated error and pitch or roll rotations.

As for the comparison of the two template correlation techniques, it can be observed that accuracy is very similar, being slightly better for the multiscale Harris. For wide rotations over 60°, single scale matching achieves the lowest error, partly because of the effect of the biggest scale features, which deal badly with wide rotations. Since the results for these two techniques is very similar, further testing has been carried out. Figure 7.10 shows a time slice of the processing of a video sequence. This processing has



(a) Re-registering and other corrections deactivated

(b) Re-registering enabled, but the coordinate of the re-registered texture is obtained from the GT

Figure 7.9: Comparison of tracking errors for different feature detectors

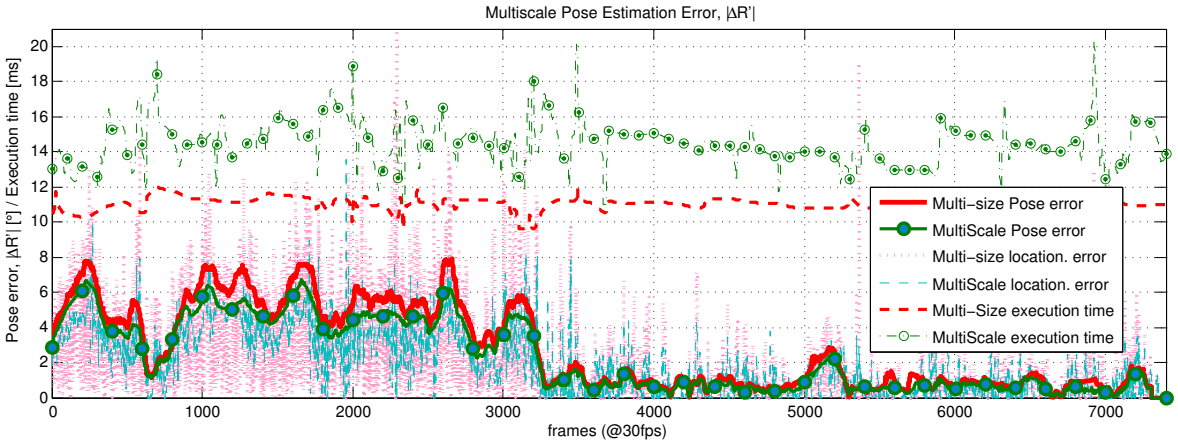


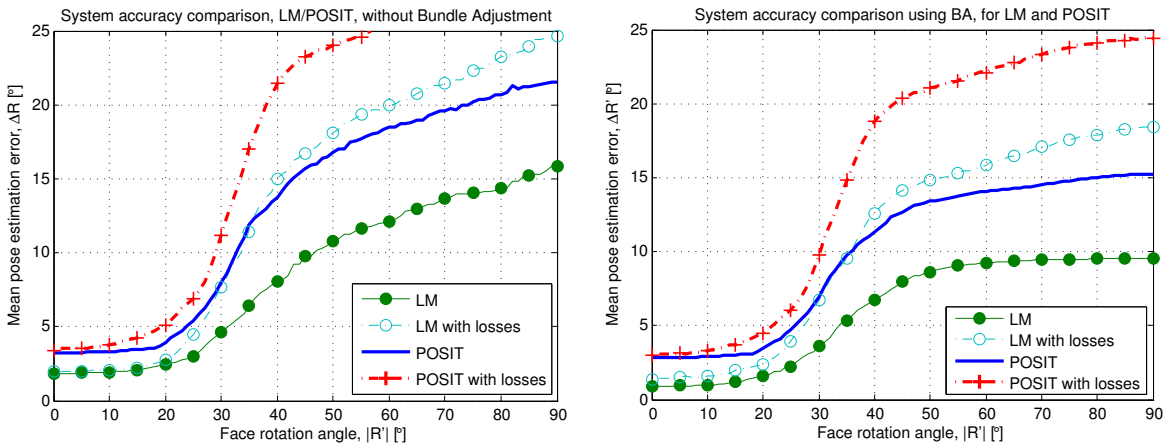
Figure 7.10: Comparison of error and execution times for the multiscale and multisize matching vs. the single-scale, multisize matching. All corrections enabled

been undertaken with all the algorithm corrections enabled, so it depicts the final pose estimation error, using multiscale tracking or multisize tracking. The error for the former is slightly bigger. But the highest difference resides in the error variation: it is much smaller for the multisize, meaning that it's more robust than multiscale. However, the execution time is also slightly bigger, because it uses bigger patches for correlation. The multiscale, on the other hand, simple requires a scale conversion operation as a previous step. With multiscale, the complete execution takes around 11 ms, while with the multisize the time rises up to 14 ms to track 30 features. Since robustness is an important parameter, the use of multisize is preferred.

7.4.4 Performance of the pose estimation with model correction

The last steps of the algorithm are the pose estimation and model correction process. Figure 7.11(a) depicts a comparison between LM and POSIT. As it was expected, LM performs better than POSIT. For both algorithms, RANSAC is allowed to run while real-time performance is not compromising. This makes an average of 18 ms, for an input image buffer of 1 second, and taking into account the time consumed by the tracking step. If there are successive frames with very low error, less RANSAC iterations are required and the image buffer empties. This allows for higher iteration time if the tracking step is producing more outliers, until the buffer is full again. POSIT needs around 1.4 μs for iteration in the given hardware platform, allowing an average of over 10K iterations. Meanwhile, LM needs around 2.7 μs, so it can do roughly 5K RANSAC iterations.

Figure 7.11(b) shows the correction effect of the underlying bundle adjustment process, which can be compared to figure 7.11(a) where BA is not applied. corrections is especially visible for yaw rotations over 30°. This is because it is the approximate rotation angle where model extension occurs. It can be observed in both graphs that error increases suddenly at that point, even when using BA, because of the addition of new points to the model also adds some error.



(a) Without bundle adjustment

(b) With bundle adjustment

Figure 7.11: Pose estimation improvement applying the BA algorithm. The error results are shown using POSIT and LM. Results are calculated with no outlier from tracking, and in case of up to 25% of outliers (losses). The outliers are errors not detected from tracking

Pose estimation error in the three direction angle rotations are shown in table 7.1. Mean error is computed by rotation ranges for each direction, as shown in the different

columns of the table.

In table 7.2, the performance of our proposal is compared with the most significant works showed in the chapter 2, State of the Art, and in table 2.2. For comparison, the full range mean errors of our proposal are also shown in the first column. Each full range mean error has been calculated as the mean of the error plot as a function of the face rotation. If the mean errors were averaged over time (average of all individual error measurements), the influence of the most common poses error would be higher in the final mean error values. Note that other authors do not specify how they calculate this error to account for the fact that the face is most of the time looking forward.

The face pose estimation system has a very low error thanks to the BA corrections. The error remains low for the full range $\pm 90°$ of yaw rotations. These results show lower error than other works in the literature, presented in chapter 2. Because the re-registering technique can not be applied under pitch variations, the system error is higher in this direction, and it can be observed how it increase for pitch angles $\alpha_{pitch} > 30°$. Still, the BA slightly improves the results. Even when it is not possible to apply re-registering for roll rotations, the patch warping works very reliably, since it is equivalent to applying a simple 2D image rotation to feature patches. Consequently, the roll error is lower than pitch error. It was not possible to evaluate the error in a wider pitch and roll range because while driving big rotations are not typical nor natural. Figures 7.12 and 7.13 depict some results for small pieces of videos. Figure 7.14 depicts a sequence of video in which the driver moves generating bright illumination in the face, and talks through a microphone to the instructors, generating occlusions and face deformation.

Rotation	$\alpha < 15°$	$15° \leq \alpha < 30°$	$30° \leq \alpha < 45°$	$\alpha >= 45^o$
yaw	1.92	2.44	6.72	12.83
***yaw* with BA**	**0.98**	**1.54**	**3.04**	**8.54**
pitch	3.82	7.86	8.59	-
***pitch* with BA**	**1.81**	**4.70**	**6.34**	-
roll	1.27	2.06	-	-
***roll* with BA**	**1.16**	**1.75**	-	-

Table 7.1: Mean face pose estimation error. The error is divided into *yaw*, *pitch* and *roll*, and evaluated in different ranges of the absolute rotation angle in the ground truth, α

Proposal	Rotation mean error			Max rotation		
	yaw	*pitch*	*roll*	*yaw*	*pitch*	*roll*
Automatic incremental 3D model	**3.8°**	**4.26°**	**1.71°**	**90°**	**45°**	**30°**
LK & online learning [Sheerman-C. 09]	4.2°	3.9°	3.1°	20°	40°	30°
3D Deformable models [Krinidis 09]	2.2°	2.6°	0.5°	60°	30°	15°
Relevance Vector Machine [Lin 09]	4.1°	2.3°	2.4°	80°	25°	10°
3D model & PF [Murphy-Ch. 08]	3.39°	4.67°	2.38°	90°	45°	45°
SIFT [Zhao 07]	2.44°	2.76°	2.86°	45°	45°	45°
Particle Filters [Oka 05]	2.86°	2.34°	0.87°	40°	20°	10°

Table 7.2: Face pose estimation error comparison with other approaches

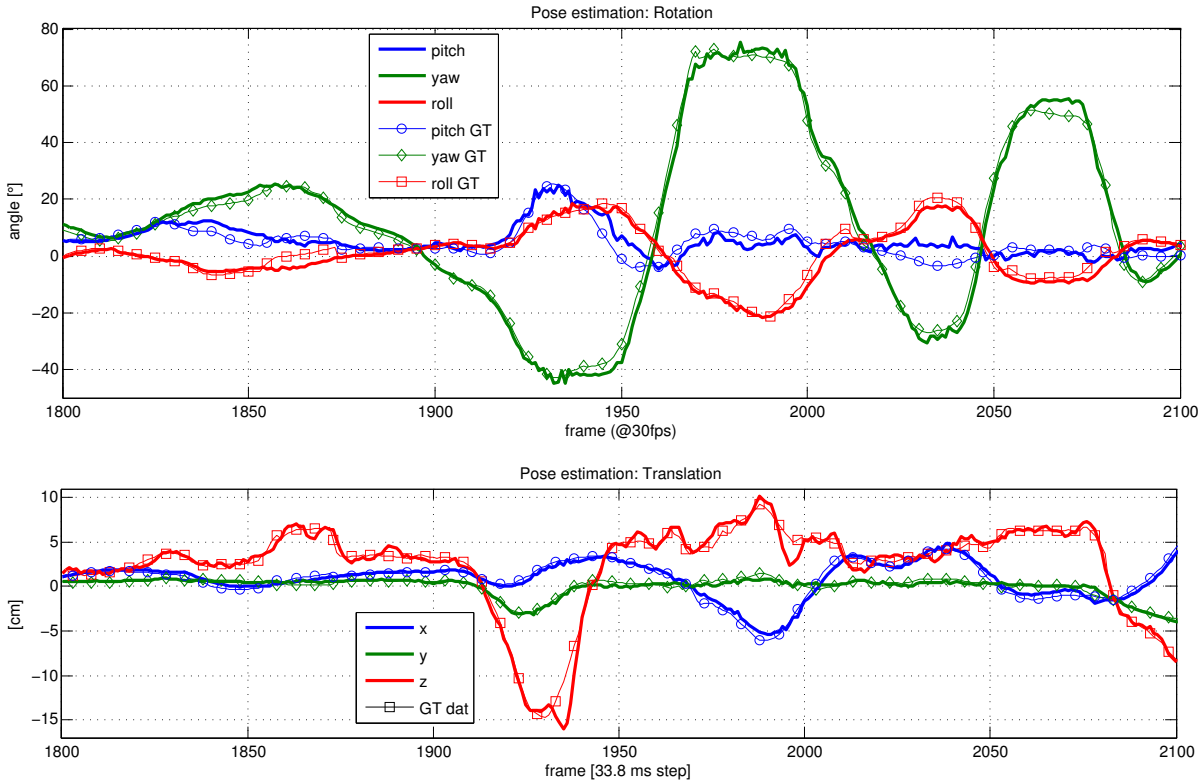


Figure 7.12: Example of pose estimation. Ground-truth data is shown for comparison

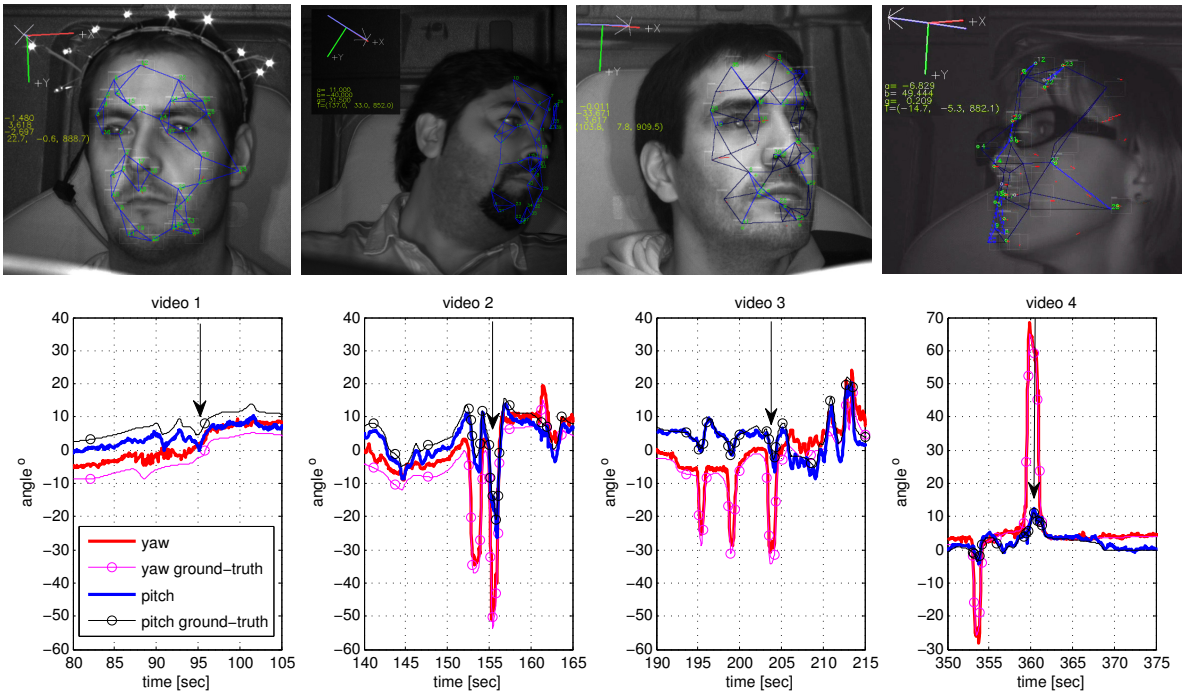
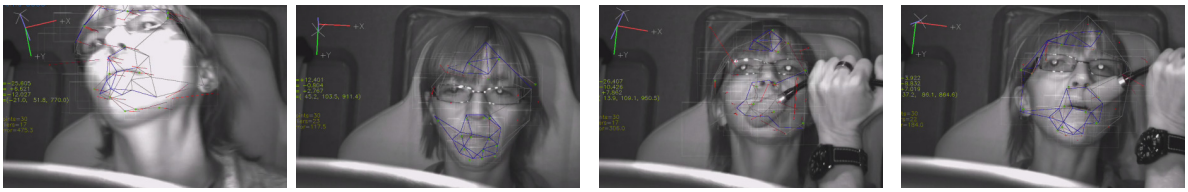


Figure 7.13: Yaw estimates over fragments of four different video sequences. Positive and negative peaks on yaw indicates when the driver is looking at the right or left mirror respectively. The fist image shows the light pattern affixed to the head, used as ground truth. The pose estimation is divided in rotation and translation. This results were obtained using the best case parameters, after the BA has optimised the 3D model.



(a) Illumination (b) Looking computer (c) Occlusions (d) Talking

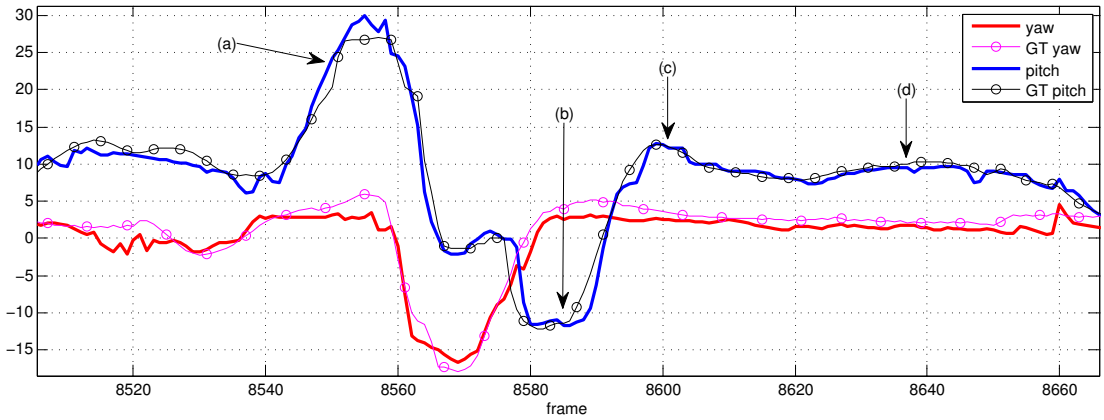


Figure 7.14: Sequence depicting illumination changes, occlusions and talking

7.5 Distraction analysis using gaze estimation

This section presents the tests and results of the non-intrusive approach to driver's gaze estimation presented in chapter 6. From this information, fixation in the scene is calculated in order to infer driver distraction state. Different distraction tasks or activities are inferred in a realistic simulator and a study of the incidence of these distracting tasks in the driver's behaviour is carried out. Experiments layout, driver's behaviour studies results and conclusions are presented.

To test the gaze estimation system, experiments where carried on the truck naturalistic simulator. There, many professional drivers were invited to drive the truck through a few scenarios carefully designed by a team of psychologists from the Safety and Human Factors Investigation and Training (ESM) centre [ESM 11], who later examine the generated data to extrapolate behaviour. The scenarios were designed and prepared to require a high level of attention from the driver, and some tasks are intentionally programmed during the driving activity to stress the driver in order to study his/her behaviour under such conditions. The simulator cabin is fully equipped with a variety of IVIS, included the face pose and gaze estimation system presented in this thesis. The gaze estimation system, along with other on-board sensors and experimental capturing systems, provides invaluable information to the psychologists who generated the experiments.

7.5.1 Experimental environment

Different aspects must be considered in the experimental environment: the camera vision system for gaze estimation, the physical simulator layout, the experiments setup, the subjects and the experiments validation. All of them have been developed under the CABINTEC project [CABINTEC 11] and supported by the MICINN (*Ministerio de Ciencia e Innovación*).

The camera vision system is described in section 7.1 at the beginning of this chapter. It is located inside the cabin, over the dashboard, between the windscreen and the driving wheel, and facing the driver. The physical simulator layout is the naturalistic truck driving simulator and covers the simulation room configuration, hardware available and geometric constraints. The experiments setup refers to all other aspects beyond the simulator layout, available hardware and information systems: the driving scenarios, road selection, inserted events while driving, schedule, etc. The experiments are formed by a set of *exercises* or *tests*, each having its own setup. These are necessary aspects to obtain good results from the exercises, and are fully designed by the team of psychologists. The subjects are the professionals who drive during the experiments, and whose behaviour is under study. Finally, whole experimental environment is validated, before starting the exercises with the subjects.

Naturalistic truck driving simulator

The experiments are accomplish in Research Facilities at CEIT [CEIT 11], San Sebastián, in a room with controlled light and sound environment.

The naturalistic simulator *TUTOR* [Lander 10], shown in figure 7.15, consists of a real truck cabin, motorised to simulate movement and equipped with common IVIS. The cabin is assembled on a movement platform with 6 degrees of freedom on which drivers can feel the vehicle accelerating, braking, its centrifugal force, etc. The devices send information to the host, located at the Instructor Position (*PI*), where the psychologists can control the whole simulator, analyse all the data and reproduce stored simulations. Main computers are placed in the PI, located behind the cabin. A dedicated computer processes face pose and gaze estimation using the algorithms presented in this dissertation, and sends this information to the PI, where the psychologists can access to the data.

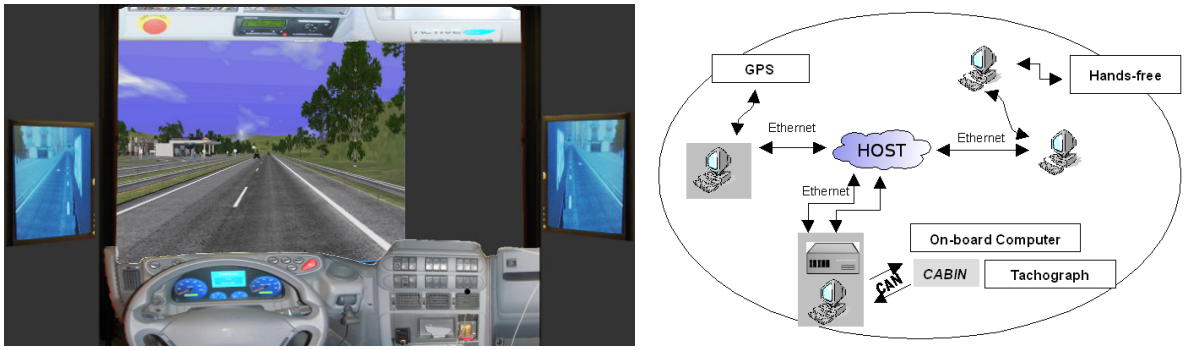
(a) Motorised simulator cabin, and projections panels (b) Instructor position (*PI*) and host computers

Figure 7.15: Naturalistic truck cabin simulator

The visualisation system is made of three back-projection panels with a total surface of 22 m^2. The fact that the screens have no marked separation plus the geometry of the image system makes for a flawless overall impression. Moreover, two monitor screens are used as rear mirrors, attached to both sides of the cabin.

The cabin is fully equipped, and contains a GPS, a hands-free, the on-board computer and a tachograph. These are some of the key locations that the gaze estimator must

differentiate. Figure 7.16 shows a representative diagram of the devices setup.



(a) View of the gaze coverage area

(b) Diagram of IVIS communications

Figure 7.16: Host and visual area setup

Experiments setup

To design the experimental protocol, the team of psychologists built on the following initial hypothesis: "The potential driver distraction due to IVIS is determined by the level of attentional demand required by them while driving, decreasing the effectiveness of the primary task: driving."

By analysing the professional drivers behaviour, the basic and most representative features in the context of this activity are identified [Kay 98]. Some scenarios, types of vehicles, incidents, on-board systems utilisation and critical situations are selected to infer distraction in drivers. Thus, the professional drivers behaviour should be generically represented. Taking into consideration this basis, which involves observing and information recording during the activity of driving, the next step is to define the basic simulation exercises.

Experiments have been designed with the goal of refuting the initial hypothesis of the research regarding the potential distraction of four different on-board systems which are commonly used in professional driving. These devices are digital tachograph, GPS, hands-free and on-board computer. Under these conditions, four scenarios have been created: mountain, inter-city, urban and long-distance. Different exercises setup have been prepared for each scenario, each containing different tasks, events, weather conditions and IVIS requirements.

According to [Victor 05], three different tasks are of special importance to study distraction: visual tasks, auditory tasks, and cognitive tasks. During the experiments, visual tasks require to use the GPS. Auditory ones can be created by making a call to the hands-free telephone, and enforcing a trivial conversation. For the last one, a cognitive task is enforced in one of the exercises by making a cognitive phone call, in which the driver is asked to describe the route from one point to another or a city she/he knows. During the exercises, the inserted events proximal to tasks, include motor, tires or ABS breakdown and other vehicles such as sudden brake of the precedent vehicle, broken down vehicles on the road, vehicles running a red light, etc. A summary of the different exercises setup is shown in table 7.3.

These tests were implemented using 16 different exercises: five of them were based on the inter-city scenario, four on the mountain scenario, three on the urban and the last four on the long-distance one. The defined procedure to evaluate these exercises consists

Scenario	Exercise	Events	IVIS
A. Inter-city	1 Control exercise	A vehicle running a STOP. Mechanical fault in air filter†‡. Cyclists on road. Sudden speed down of preceding vehicle. Slow vehicle on road.	GPS Hands-free
	2 GPS guidance		
	3 Faulty GPS guidance		
	4 Telephone guidance		
	5 GPS guidance Distorted voice call		
B. Mountain	1 Control exercise	Obstacle on road A vehicle running a STOP. A vehicle stopped on road. Sudden speed down of preceding vehicle. Slow vehicle on road. Tyre blowout‡.	GPS Hands-free Tachograph
	2 GPS guidance		
	3 Telephone guidance		
	4 GPS guidance Faulty voice call Tachograph speed warning		
C. Urban	1 Control exercise	ABS fault† A vehicle running red light. Mechanical fault in air filter†‡. A pedestrian crossing the street. A dog crossing the street.	GPS Hands-free Tachograph Computer
	2 GPS guidance tachograph error		
	3 GPS Guidance Distorted voice assistance call		
D. Long-distance	1 Control exercise	Obstacle on road A vehicle running a STOP. Vehicles stopped on the road. Slow vehicle on road, and a car overtaking a bus downhill.	Hands-free Tachograph Computer
	2 Phone calls On-board computer warnings		
	3 Phone call Cognitive phone call* Tachograph warnings		
	4 Phone call On-board computer data		

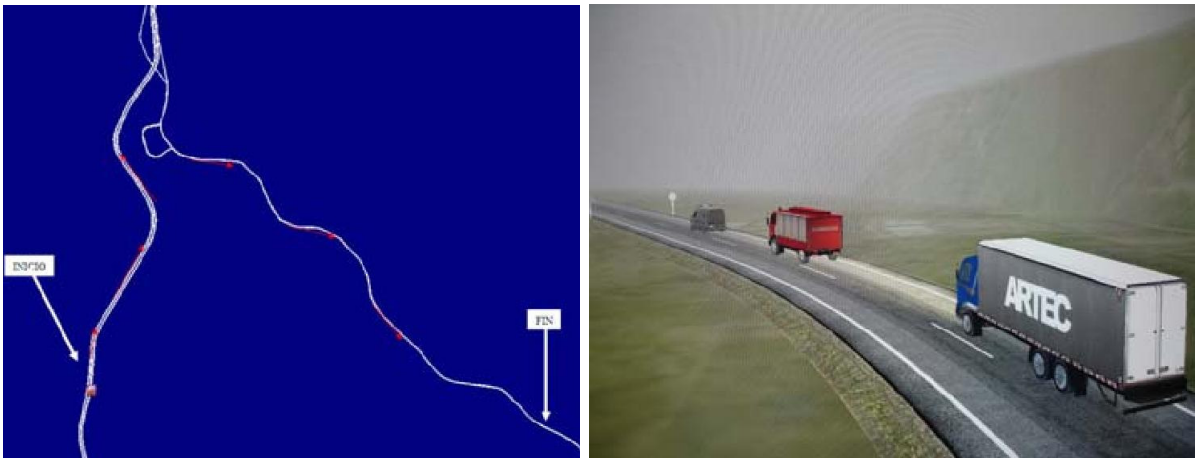
†Marked on the on-board computer
‡Truck dynamic model changes
*Phone call with important cognitive charge: The driver is asked to explain a route within a known city.

Table 7.3: Exercises setup

on different drivers driving through different scenarios.

The first exercise of each scenario is the "Control exercise" which corresponds to the exercise undertaken by each driver in the different scenarios without external perturbations. It is important to have these control exercises, because using them, the behaviour at different points of the scenarios in the subsequent tests of each driver can be compared and correlated with the distraction sources. Once the chain of exercises is finished, enough information is provided about the drivers behaviour while driving in order to generate a distraction pattern for each one.

The Figure 7.17 depicts the scenario and a view of the truck for the exercise B.2.



(a) Exercise scenario (b) Example of the view of the truck from the PI

Figure 7.17: Exercise B.2

Subjects

According to the previous considerations, the number of tests and their configuration, a minimum number of participants of 12 was set, in order to have one participant for each test configuration to detect the dependent behaviour variables.

It is important to highlight that every participant needs to pass a test to exclude people with propensity to suffer simulator-sickness. Previous studies with similar conditions used

Exercises:		Inter-city					Mountain				Urban			Long-distance			
		A.1	A.2	A.3	A.4	A.5	B.1	B.2	B.3	B.4	C.1	C.2	C.3	D.1	D.2	D.3	D.4
Subjects:	1	x	x				x	x			x	x		x	x		
	2	x		x			x		x		x		x	x		x	
	3	x			x		x			x	x	x		x			x
	4	x				x	x	x			x		x	x	x		
	5	x	x				x		x		x	x		x		x	
	6	x		x			x			x	x		x	x			x
	7	x			x		x	x			x	x		x	x		
	8	x				x	x		x		x		x	x		x	
	9	x	x				x			x	x	x		x			x
	10	x		x			x	x			x		x	x	x		
	11	x			x		x		x		x	x		x		x	
	12	x				x	x			x	x		x	x			x

Table 7.4: Test configuration

groups from 7 to 30 participants [Ting 08, Lee 07a]. All subjects were informed of the purpose of the experiment and the security procedures in the simulator facilities. Table 7.4 depicts the subject and exercise relation for each test configuration.

Experiment validation

The experimental model is validated following a three-stage strategy:

- *STAGE 1*: A group of drivers with information about the technological tools and distraction sources while driving. In this stage, the exercises and experiments have been designed based on the analysis of tasks and taking into account the objectives and the underlying assumptions of the investigation. The main objective of this stage is the validation of the designed tests.
- *STAGE 2*: Professional drivers group. At this stage a group of 5 professional drivers who know the objectives of the research and the simulation environment advise the researchers to improve the exercises and tests to set up a simulation environment that feels more realistic to them.
- *STAGE 3*: Final drivers group. A representative sample of drivers are selected for this group. It is composed of at least 12 drivers from different gender, age, experience, etc to obtain conclusions about distraction and driver's behaviour.

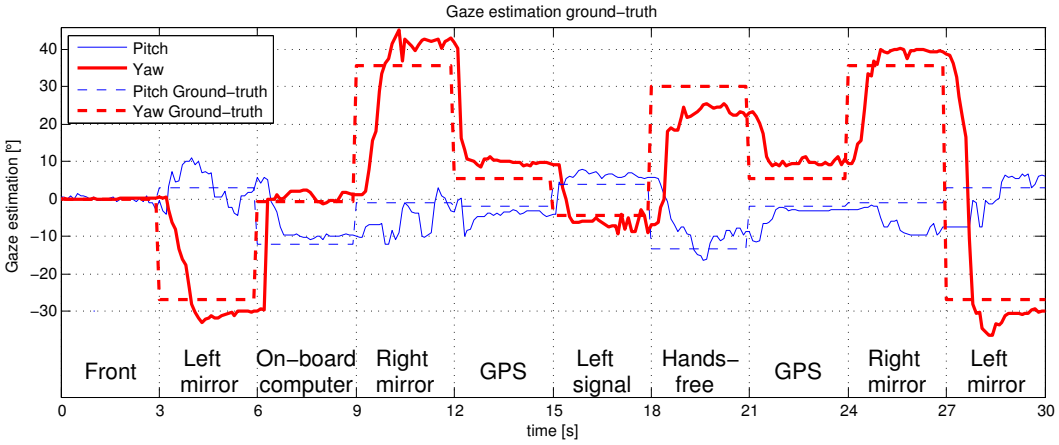
7.5.2 Gaze estimation performance evaluation

Prior to further analysis on the data obtained by the gaze estimation system, the researchers and psychologists who are going to work with these data require an assessment of its reliability and accuracy. Obtaining ground-truth data for these experiments is more complex than for the face pose. As a coarse ground-truth, we have used the simulator cabin itself and a pattern image projected on the frontal projection panel, as shown on figure 7.1(b). A voice record asks the subject on the driver's seat to consecutively look at different key-points on the simulator and screen, in a random sequence. Synchronously, the video system records a video and annotates the locations where the user is asked to look at. Since the geometry of the simulator is known, it is possible to calculate the angles at which each key-point is located, and compare these positions with those resulting from the gaze estimation of the videos sequences.

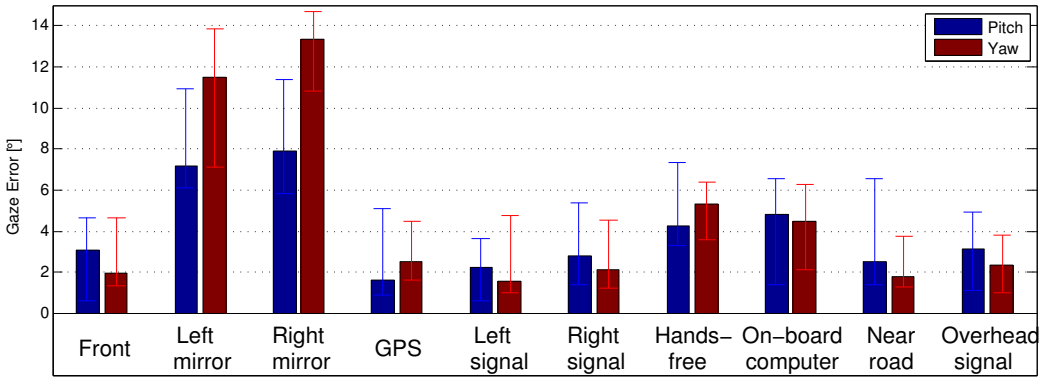
This procedure does not take into account the reaction time and the small variations on fixation that any subject experiments during focalisation and during extended fixation periods. For this reason, the user is asked to look at each key-point during three seconds, but only one second is averaged to compute fixation and compare it with the ground-truth data. Figure 7.18(a) shows a test gaze pattern compared to its GT for one of the user. This procedure was repeated by four subjects, and the average error for each key-point is depicted in figure 7.18(b).

7.5.3 Gaze estimation results

The gaze estimation system proposed in chapter 6 is evaluated using the resources available in the simulator, placed in the CEIT Research Facility, and following the experimental environment described above. Generating a complete report of the results of the tests accomplished in the simulator is out of the scope of this thesis. Thus, detailed results for



(a) Gaze estimation compared to a ground-truth sequence



(b) Mean gaze estimation error

Figure 7.18: Gaze and focusing estimation error

one test are presented in this section, and summary tables will contain statistics for all of the tests.

Figure 7.19 shows a view of the gaze estimation data and fixation classification as it is visualised by the psychologists. It depicts a driving sequence on exercise D.2, at a moment where a few events collide, requiring a high degree of interaction to the driver. The graphic represents the classification of the area of attention based on the gaze estimation.

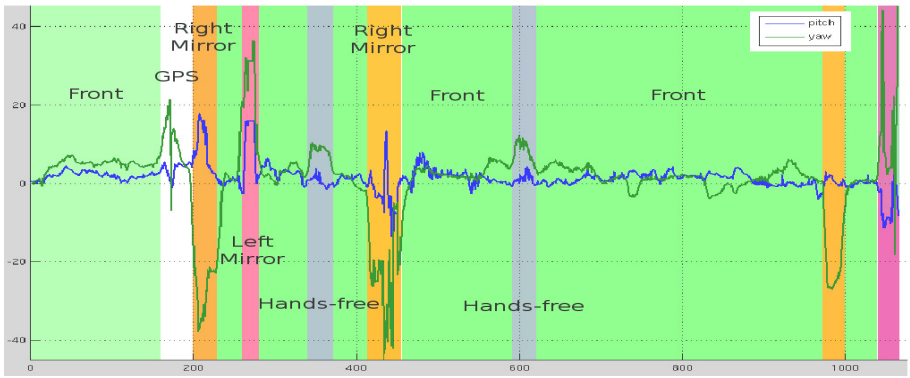


Figure 7.19: Driver's gaze estimation and fixation areas classification during driving

Figure 7.20 shows the moment on exercise D2 when three events intentionally collide.

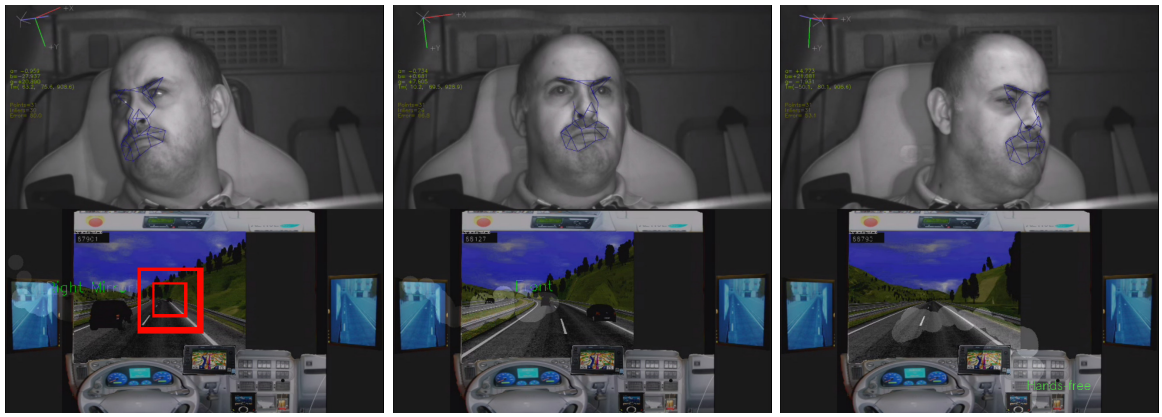
A car appears stopped on the right margin of the road, and at the same time, another vehicle is overtaking the truck. The figure clearly depicts how the subject repeatedly looks up to five times to the left mirror first, before changing to the other lane, how he follows with his eyes the passing car, and how he looks to the right mirror to come back to the right lane. After passing the stopped car, the subject receives a phone call through the hands-free device. This sequence can help the psychologists understand and study the behaviour of the driver under this stressful situation.

The graphic on figure 7.20 show that the subject needed 7 seconds to realise that a car ahead on the road is actually stopped and it is not just a slow vehicle. The driver previously had simply checked the speed to slow down. At this moment, the driver starts the needed actions to overcome the obstacle (looking at the mirror to overtake the obstacle). Later on, he receives a phone call, but being busy, he takes up to 10 seconds to answer. While taking to the telephone, he loses attention to the road for at least 4 seconds, even though he is using a hands-free. As for figure 7.21, it shows that the roadwork operations generate a high level of distraction to the driver, who overpasses speed limit when approaching the round-about at the end of the road.

After the experiments, a similar comprehensive analysis was done over the whole video dataset recorded at the simulator to extrapolate the subjects' behaviour from the gaze and other data generated by the simulations. Table 7.5 shows these results. It compares the reaction times needed before or after an inserted event in the case of the control exercise or when the subject is forced to a distractive driving by requiring a high degree of devices utilisation. Each scenario has a set of scheduled events, as shown in table 7.3, so an undistracted driving (in the control exercise) can be compared to the other exercises, which add GPS and phone distractions on the same journey than the control one. Table 7.5 shows some important statistics inferred after the complete evaluation of the gaze for all the subjects and exercises. As an example, on exercise D3, few drivers are unable to avoid hitting an on-road obstacle, while they perfectly do it up to 12 seconds in advance if they were not distracted. Many of them overpass speed limits more often, and need more time to notice a mechanical failure. One of the subjects needs more than two minutes to notice that he is driving a fully loaded truck on a mountain road with a flat tyre. Being undistracted, he needed a few seconds to notice the same anomaly. The gaze estimation system allows to study what was the subject doing before noticing the anomaly, and why he wasn't aware of that for such a long period.

To give a better understanding of the distraction pattern of a driver, two more different ways of measuring distraction can be applied, apart from reaction time. Traditionally, a glance-based measure has been used. This measures the duration of individual fixations on different areas, the frequency, number of glances or total task duration. However, this measurements heavily depends on the task, the driver experience and other factors. [Victor 05] found that the *Percent Road Centre* (PRC) measurement is much more stable across users and different experiments. PRC measures how much time is spent monitoring the road centre area while performing a task. This area includes the road, signalling and other visual elements proximal to the road. We have analysed these parameters in our experiments.

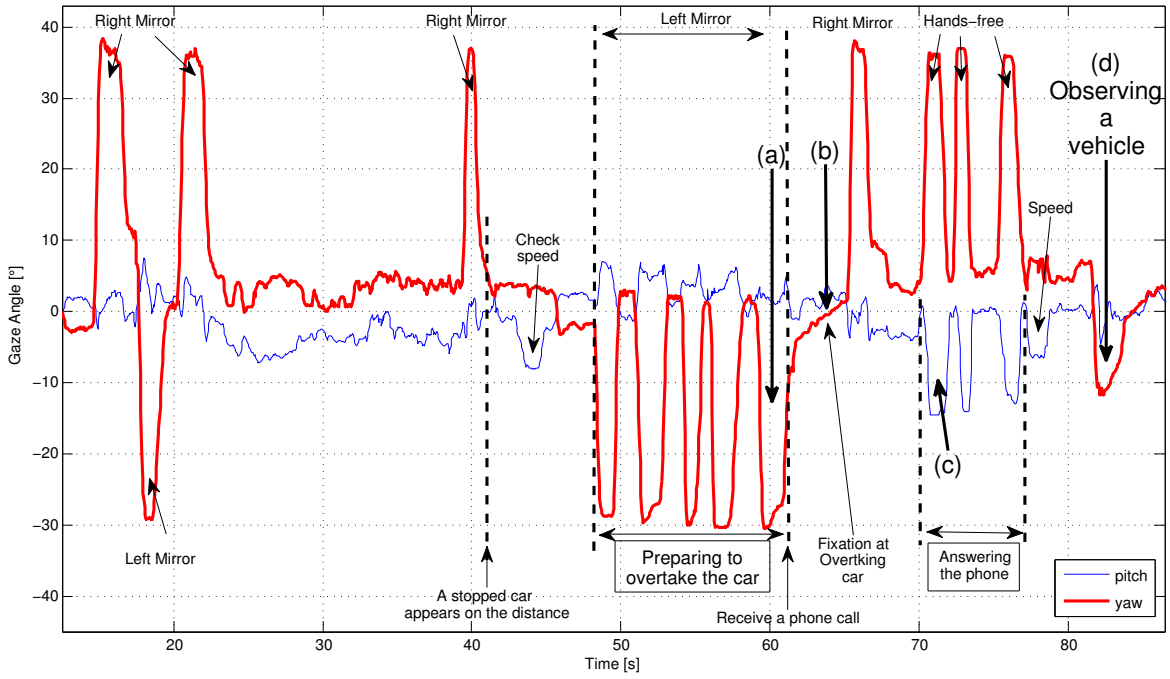
Figure 7.22 shows the focusing time percents during the execution of a tasks. This figure was generated after the analysis of the results obtained with the gaze estimation algorithm and tasks schedule during the exercises. The first column shows the average percents for the control experiments (exercises A1, B1, C1 and D1). The second column depicts visual distraction using the GPS, obtained from exercises A2, A3, B2, B4 and C2. The third column depicts auditory distractions inferred by tasks requiring talking

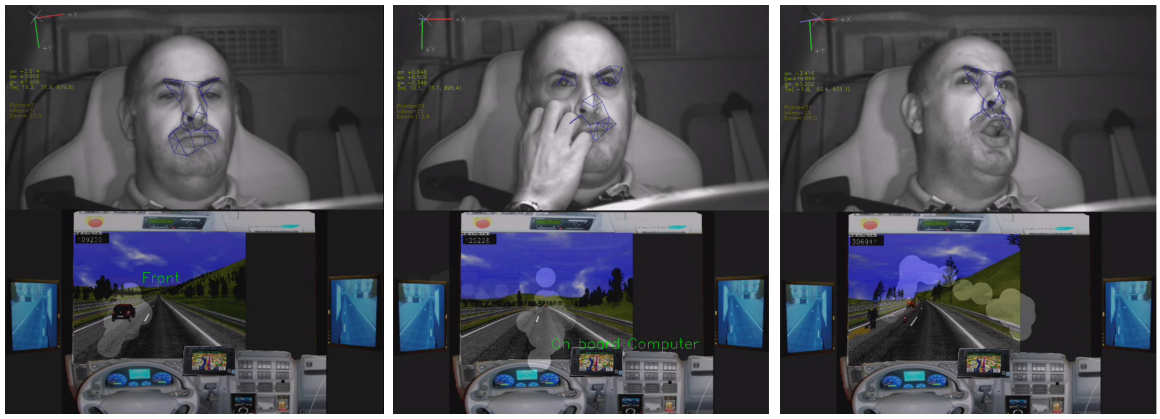


(a) The drivers is trying to change to the other lane

(b) Overtaking the stopped car

(c) Overtaking the stopped car





(d) Overtaking the stopped car

(e) Partial occlusions

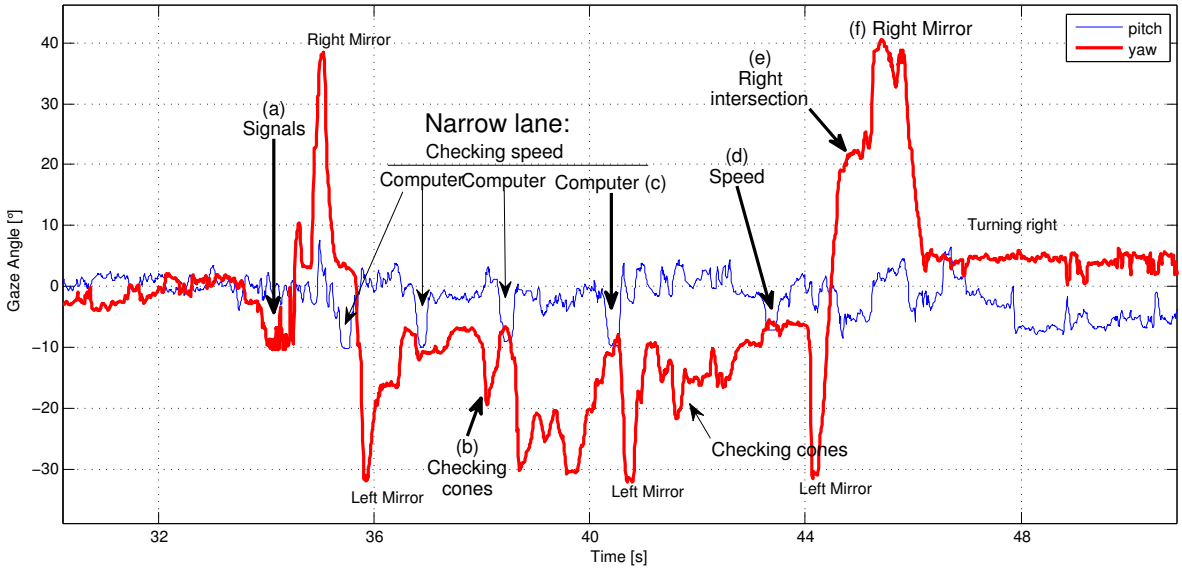
(f) Talking

Figure 7.20: Exercise D2. Sequence of the video in which a car is stopped on the road, but another vehicle is overtaking the truck. Then the driver receives a phone call

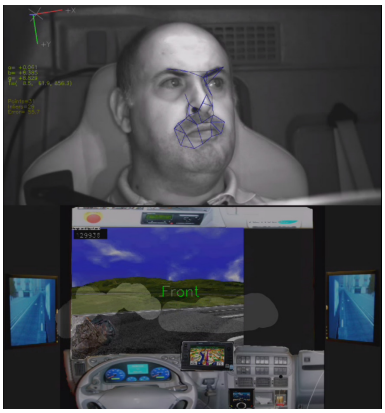
(a) Roadwork area: Checking visualising signals

(b) Observing the roadworks and lane space

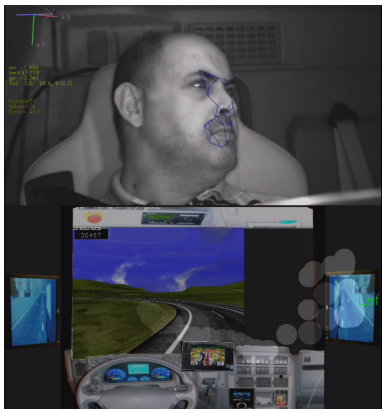
(c) Checking speed on the on-board computer



(d) Intersection: Checking speed



(e) Intersection: Looking right



(f) Turning: Checking right mirror

Figure 7.21: Exercise D2. Coming into an roadworks area, and a roundabout at the end of the road

Scenario	Exercise	Overpass speed limit [#]	Reaction time to an event [seconds]					
			Obstacle‡		Mechanical fault*		Answer a call*	
			max	min	min	max	min	max
Inter-city	A1	0	25	5	(1) 5	9	1.5	5
	A[2-5]	2	15	3	32	81	2	9
Mountain	B1	2	19	5	(2) 0.6	2	1	3
	B[2-4]	6	13	2	24	2min 11s	2	11
Urban	C1	0	16	3	(3) 4	11	2	8
	C[2-3]	0	4	0 †	4	43	4	miss
Long-Distance	D1	1	34	12	(4)	-	1	4
	D[2-4]	4	20	0		-	3	miss

* Reaction time after the event.
‡ Reaction time before the event. Moment at which the subject is aware of the obstacle and takes an action before colliding. (The higher the better)
† Do not have time to react before hitting the obstacle. Collision produced.
(1) Mechanical fault in the air filter. Warning marked on the on-board computer, smoke visible in rear mirror, and truck dynamic model changes.
(2) Tyre blowout. Marked trough audible sound and truck dynamic model changes.
(3) ABS fault. Warning marked on the on-board computer.
(4) No mechanical faults scheduled.

Table 7.5: Driver behaviour and reaction time statistics

by the hands-free phone, on exercises A4, A5, B3, D2 and D4. The last one represents a cognitive task on exercise D3, which induced distraction with a phone call to explain a route.

We have found PRC to be very correlated with the level of distraction of the driver. Moreover, we also found that the characteristic pattern of visual, auditory and cognitive distractions ones are different. While the former shows an important reduction of PRC, auditory tasks does not reduce PRC, rather instead, it slightly increases. On cognitive tasks, we could not infer any important variation of this parameter. However, the time used looking at the signalling and road proximities is reduced for all tasks. This behaviour is clearly observable on figure 7.22, and it is in line with the conclusions presented by [Victor 05].

It can be observed how the time that the driver spends looking at the mirrors, signals and on-board computer is drastically reduced for any of the task, in comparison with the control exercises. During auditory tasks, the drivers increases the time in which he/she is looking to the front, but reduces the fixations on the mirrors and signals. This means that although the driver is looking forward, he/she is not paying attention to the rest of the scene, as a normal responsible driving would require.

7.6 Conclusions

In this chapter we have evaluated the performance of the face pose and gaze estimation approaches. The comparison of the video processing results with the ground-truth data shows that the error of the face pose estimation is fairly good, showing better results than

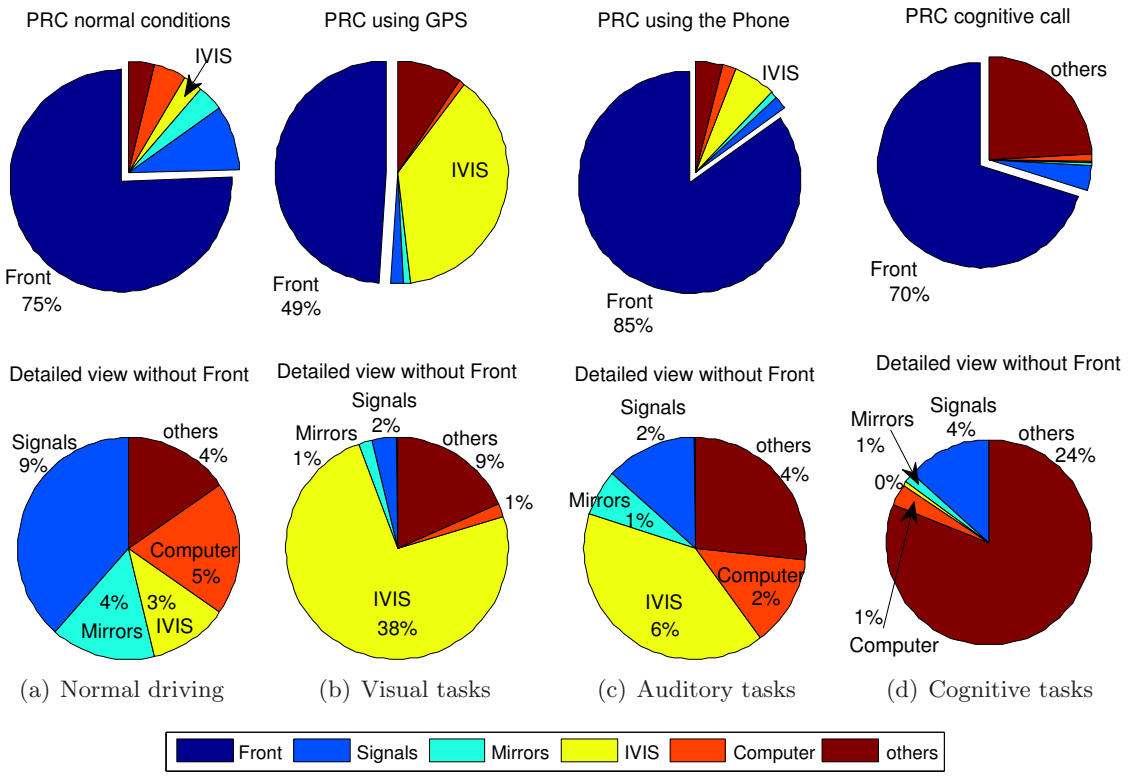


Figure 7.22: PRC statistics

other approaches presented in the State of the art. It is especially remarkable the low error that the algorithm presents under yaw rotations, being a full-range and full-automatic system. This demonstrates that the re-registering process, the model extensions and the bundle adjustment processes collaborate well to produce a robust and accurate estimation. The error estimation is obtained for the typical range of movement presented by a driver on the seat.

After reviewing the results presented, the user of the face pose estimation system can choose a more accurate estimation using the LM algorithm, or a faster approach using the POSIT algorithm. Both solutions present good accuracy, robustness and full range estimation. The algorithm to use would depend on the system requirements.

In addition to the pose, the gaze gives a very valuable information about the driver behaviour. Using the gaze and focusing estimation from the videos recorded in the *TU-TOR* simulator, it was also possible to study how the distraction patterns induced to the drivers changes the distribution and duration of focalisation periods on the different devices and how this affects their reaction times. The analysis of this information by a team of psychologists can help to improve driving teaching and safety.

Chapter 8

Conclusions and Future works

Driver distraction is one of the main causes of traffic accidents, which cost many lives and money every year, everywhere in the world. It is known as one of the main causes of dead in young people. Many of these accidents happen only few seconds after the distraction started, and could have been avoided easily if the driver were paying attention to the road. At this point is where an automatic monitoring system can help reducing the number of accidents. Driver monitoring is a complex task, and involves many parameters of behaviour and physiology. Analysing head movements, facial expressions and actions like blinking or gaze fixation using computer vision can help to estimate the state of attention of the driver.

The objectives of this thesis are to create, using computer vision, a face pose and a gaze estimation algorithm, to assert the reliability of the 3D model creation process, the multisize matching and the re-registering technique, and to test the proposed algorithm in a simulator to study a person distraction behaviour while driving. These studies provide very valuable information to instructors and professional drivers, and will help for a better understanding of the distraction sources inside a vehicle. In addition, this algorithm can also be used on-board of utility vehicles, where the system can rise a warning when it detects that the driver has not been paying attention to the road for a relatively long period.

We have implemented a real-time 3D face pose estimation system. The proposed algorithm is a fully-automatic and user-independent system based on a set of face features. The only calibration required is the stereo camera rig calibration, which is done offline. A sparse 3D model is automatically created during the first frames of execution, but a technique to refine and improve the model during the whole execution time has been evaluated. This model creation extended in time generates a very accurate model of the face of the subject, providing a very precise pose estimation. In this sense, we have evaluated how to initially create a good model and how to extend it with new features of the face. Adding new points to an existing model at a given pose introduces error in the model, because the error in the estimated pose at the moment of addition is transmitted to the new points, and consequently degrades the whole pose estimation. To overcome this problem, a bundle adjustment algorithm [Triggs 99] has been tested to correct the model after points addition. The final result is an accurate sparse 3D model with very low error.

In this thesis, we have studied different methods to extract adequate face features and establish stereo correspondences. We found that the well-known and extensively used SURF [Bay 08], does not provide good results due to the lack of irregularities and corners in the face, and the low ambient illumination. Instead, we implemented a novel *multisize*

matching technique, based on Harris interest points [Harris 88] and patch correlation. This technique joins the goodness of different patch sizes for correlation. Smaller patches give better performance under rotations, while being less sensitive to illumination changes. Bigger ones, on the other hand, are more robust although less accurate.

A typical problem which limits the rotation range of many pose estimation systems is how to deal with the changing appearance of a feature under 3D rotations. We implemented a new re-registering technique which takes advantage of the stereo cameras disposition, and allows for a full range and very accurate face tracking from -90° to +90° yaw rotations. In this technique, as the face rotates, we use the forward camera in the direction of rotation to capture new texture patches of the features, and the backward camera to track using the patches that were previously captured. This means that a texture patch needs to be tracked only for a range of ±7.5°. For pitch and roll rotation, where the developed re-registering technique can not be applied, we tested patch warping.

The system has been evaluated under low light conditions and proved good results. Even so, results for the proposed pose estimation algorithm show a yaw mean rotation error below 1° for rotations in the ±15° range, and 1.54° in the ±30° range, improving the results of other works in the literature.

The face pose estimation is enhanced with a coarse eye direction to obtain the gaze estimation. The gaze estimation system has been extensively tested in a driving simulator [Lander 10] and used by a team of psychologists experts on driving behavioural, to evaluate the distraction pattern of subjects under some stressing conditions. The experiments accomplished in the simulator have provided an extensive video dataset with more than 10 users and hours of driving. These videos and a ground-truth have been used to test and evaluate the system performance.

The algorithm proved an adequate classification of the focusing area. This information, together with the driving parameters provided by the simulator are used by psychologist to obtain important results about how the utilisation of *In Vehicles Information Systems* (IVIS) affects the distraction pattern of the drivers. The statistic obtained show how reaction times increase, and gaze fixation patterns change according to visual, auditory and cognitive distractions. In conclusion, distraction caused by IVIS can become an important hazard while driving.

8.1 Main Contributions

From the results obtained in previous chapters, we consider that the main contributions of this thesis are the following:

1. **Automatic and incremental 3D face model.** A method to create a fully-automatic coarse 3D face model has been presented. The model is built incrementally, with the addition of initially occluded parts of the face, and with corrections during execution.

2. **Feature matching techniques.** The performance of different matching techniques, with the specific challenges that a face and low light conditions present, has been evaluated. As a result, SURF and patch correlation have been discarded, and a novel multisize patch correlation has been implemented.

3. **Full range rotation.** To cope with the problem of appearance changes due to rotation, a new re-registering technique has been implemented in order to obtain

a feature robust tracking. Using this technique, the pose estimation is robust and accurate for full range yaw rotations, in the range of $\pm 90°$.

4. **Face pose estimation.** Two different algorithms for 3D pose estimation have been tested. Levenberg-Marquardt gives very good accuracy, whereas POSIT is faster. A RANSAC algorithm and the usage of the 3D face model permit outliers detection and robust pose estimation.

5. **Classification of focus of attention.** Fixation areas classification system based on the gaze allows to detect the focusing or fixation point of the drivers withing a set of possible focusing areas previously known.

6. **Distraction behaviour.** Using the gaze estimation system, the distraction pattern and inattention statistics have been computed for a set of experiments held in a driving simulator. Some visual, auditory and cognitive distractions have been analysed. Parameters as reaction time and gaze fixation patterns over time have been identified as interesting for distraction analysis.

8.2 Future work

From the results and conclusions of the present work, several lines of work can be proposed.

- **Better initial feature extraction.** Although the initially created 3D face model and the online corrections proved to generate an accurate 3D pose estimation, a more comprehensive feature extraction process can help improve the results. More comprehensive means than the extracted features are associated with a meaning, that is, to know if a feature belongs to the nose, the eye, the mouse, etc. This can help treating each feature in a different way depending on their associated information, and consequently improving algorithm aspects such as occlusions, optimum patch size, calculating the pseudo-normals, or non-rigid deformations. This would produce an even better and more robust face pose estimation.

- **Deformable models.** A deformable 3D model can adjust to temporal 3D face structure variations, and adapt to gestures or speaking. By implementing a deformable 3D face model the pose accuracy could improve in these cases. Providing more information about the face features, and a dynamic model of the face would help on the task of implementing a non-rigid model pose estimation. Great care must be taken to prevent the degeneration of the model and to obtain a real time application.

- **Feature occlusions.** A cylindrical model is used to pre-calculate the occlusion angles of each feature. However, this information could also be updated online, by observing when the different feature points occlude. This requires implementing a method to detect when a feature has been occluded by other object or whether it is a self-occlusion. This could be done by similarity techniques, such as local histograms or optical flow, and a consensus though repeatability along few frames, and neighbour features and rotation angles.

- **Texture warping.** A texture warping technique is applied in this thesis. The test results showed that it only improves feature localisation for a small range of rotation, mainly because of incorrect calculation of the feature 3D orientation within the face.

Improving this calculation would allow to apply warping for a wider range, what would increase accuracy and range for pitch and roll rotations.

- **Improved calibration.** An accurate stereo rig calibration improves the 3D point estimation, model accuracy and feature tracking. In many cases the camera calibration can also be corrected with bundle adjustment, which is a common technique applied in SLAM problems. The bundle adjustment could be applied to correct calibration parameters in addition to the 3D structure of the model.

- **Better eye direction estimation.** In this thesis we implemented a coarse eye direction estimation algorithm, as a mean to provide driver's gaze and calculate focalisation. The eye direction estimation is not so accurate as the pose estimation, and consequently degrades the final gaze. A necessary step to increase the gaze accuracy to the levels of the pose estimation would be to implement an accurate eye direction estimator using the two high resolution stereo cameras.

- **Single camera face pose estimation system.** The re-registering technique developed in this thesis could work with a single camera, although with less reliability. This re-registering could be paired with a 3D model built with a structure-from-motion algorithm [Paladini 10] to create a mono-camera solution able to work in the full yaw rotation range.

- **Distraction parameters.** More effort would be needed in the identification of new parameters calculated from the gaze direction and focalisation to help the psychologists to detect distractions in an easy way.

- **Inattention monitoring system.** With all the information provided by the pose and gaze, and taking advantage of the high resolution cameras and the eye analysis accomplished for eye direction estimation, it would be possible to implement as well a driver inattention detector on top of the gaze estimator. This would provide comprehensive information about the inattention state (drowsiness and distractions) of the driver.

Bibliography

[Agilent 99] Agilent Technologies, Inc. *Application Note 1118: Compliance of Infrared Communication Products to IEC 825-1 and CENELEC EN 60825-1*, 1999.

[Alcantarilla 08] P.F. Alcantarilla, L.M. Bergasa, P. Jiménez, M.A. Sotelo, I. Parra, D. Fernández & S.S. Mayoral. *Night Time Vehicle Detection for Driving Assistance*. In IEEE Intelligent Vehicles Symposium (IV), 2008.

[An 08] K.H. An & M.J. Chung. *3d head tracking and pose-robust 2d texture map-based face recognition using a simple ellipsoid model*. In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pages 307–312. IEEE, 2008.

[Angell 06] L Angell, J Auflick, P Austria, D Kochhar, L Tijerina, W Biever, T Diptiman, J Hodgsett & S Kiger. *Driver workload metrics project: Task 2 final report*. Rapport technique, NHTSA, 2006.

[Apostolo 02] Nicholas Apostolo & Alexander Zelinsky. *Vision In and Out of Vehicles: Integrated Driver and Road Scene Monitoring*. In International Journal of Robotics Research, pages 5–28. Springer-Verlag, 2002.

[Artac 02] M. Artac, M. Jogan & A. Leonardis. *Incremental PCA for On-Line Visual Learning and Recognition*. In International Conference and Pattern Recognition, volume 16, pages 781–784, 2002.

[Ba 04] S.O. Ba & J.M. Odobez. *A probabilistic framework for joint head tracking and pose estimation*. Pattern Recognition, vol. 4, pages 264–267, 2004.

[Baker 04a] Simon Baker & Iain Matthews. *Lucas-Kanade 20 Years On: A Unifying Framework*. International Journal of Computer Vision, vol. 56, no. 3, pages 221–255, March 2004.

[Baker 04b] Simon Baker, Iain Matthews, Jing Xiao, Ralph Gross, Takeo Kanade & Takahiro Ishikawa. *Real-Time Non-Rigid Driver Head Tracking for Driver Mental State Estimation*. In 11th World Congress on Intelligent Transportation Systems, October 2004.

[Balasubrama. 07] V.N. Balasubramanian, Jieping Ye & S. Panchanathan. *Biased Manifold Embedding: A Framework for Person-Independent Head Pose Estimation*. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1 –7, June 2007.

[Balasubrama. 09] Vineeth Balasubramanian & Sethuraman Panchanathan. *Biased Manifold Embedding: Supervised Isomap for Person-Independent Head Pose Estimation*. In Computer Vision and Computer Graphics. Theory and Applications, volume 21 of *Communications in Computer and Information Science*, pages 177–188. Springer Berlin Heidelberg, 2009.

[Baumberg 00] A. Baumberg. *Reliable feature matching across widely separated views*. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), page 1774. Published by the IEEE Computer Society, 2000.

[Bay 06] Herbert Bay, Tinne Tuytelaars & Luc Vaan Gool. *SURF: Speeded Up Robust Features*. In Eur. Conf. on Computer Vision (ECCV), 2006.

[Bay 08] Herbert Bay, Andreas Ess, Tinne Tuytelaars & Luc Van Gool. *SURF: Speeded Up Robust Features*. Computer Vision and Image Understanding, vol. 110, no. 3, pages 346–359, 2008.

[Bergasa 06] Luis M. Bergasa, J. Nuevo, Miguel A. Sotelo, R. Barea & E. López. *Real-Time System for Monitoring Driver Vigilance*. IEEE Trans. on Intelligent Transportation Systems, vol. 7, no. 1, pages 1524–1538, March 2006.

[Berka 07] Chris Berka, Daniel J. Levendowski, Michelle N. Lumicao, Alan Yau, Gene Davis, Vladimir T. Zivkovic, Richard E. Olmstead, Patrice D. Tremoulet & Patrick L. Craven. *EEG Correlates of Task Engagement and Mental Workload in Vigilance, Learning, and Memory Tasks*. Aviation, space, and environmental medicine, May 2007.

[Beymer 94] D.J. Beymer. *Face recognition under varying pose*. In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on, pages 756–761, June 1994.

[Blaschke 09] C. Blaschke, F. Breyer, B. Frber, J. Freyer & R. Limbacher. *Driver distraction based lane-keeping assistance*. Transportation Research Part F: Traffic Psychology and Behaviour, vol. 12, no. 4, pages 288–299, 2009.

[Bouguet 10] J.Y. Bouguet. *Documentation: Camera Calibration Toolbox for Matlab*. [web page] `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`, 2010.

[Bradski 08] G. Bradski & A. Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, 2008.

[Brandt 04] T. Brandt, R. Stemmer & A. Rakotonirainy. *Affordable visual driver monitoring system for fatigue and monotony*. In Systems, Man and Cybernetics, 2004 IEEE International Conference on, volume 7, pages 6451 – 6456 vol.7, October 2004.

[Brown 92] L.G. Brown. *A survey of image registration techniques*. ACM computing surveys (CSUR), vol. 24, no. 4, pages 325–376, 1992.

[Brown 02] Lisa M. Brown & Ying-Li Tran. *Comparative Study of Coarse Head Pose Estimation*. Motion and Video Computing, IEEE Workshop on, vol. 0, page 125, 2002.

[CABINTEC 11] CABINTEC. [web page] `www.cabintec.net/`, 2011.

[Canton-F. 07] C. Canton-Ferrer, J. R. Casas & M. Pardàs. *Head Pose Detection based on Fusion of Multiple Viewpoint Information*. Multimodal Technologies for Perception of Humans, Int'l. Workshop Classification of Events Activities and Relationships, CLEAR 2006, 2007.

[Carsten 05] Oliver Carsten & Karel Brookhuis. *Issues arising from the HASTE experiments*. Transportation Research Part F: Traffic Psychology and Behaviour, vol. 8, no. 2, pages 191 – 196, 2005.

[CEIT 11] CEIT. *Centre of Studies and Technical Research of Gipuzkoa*. [web page] `http://www.ceit.es/`, 2011.

[Cootes 95] T.F. Cootes, C.J. Taylor, D.H. Cooper & J. Graham. *Active Shape Models-Their Training and Application*. Computer Vision and Image Understanding, vol. 61, no. 1, pages 38–59, 1995.

[Cootes 01a] T. F. Cootes, G. J. Edwards & C. J. Taylor. *Active appearance models*. IEEE Trans. Pattern Anal. Machine Intell., vol. 23, pages 681–685, January 2001.

[Cootes 01b] T. F. Cootes & C. J. Taylor. *On representing edge structure for model matching*. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 1114–1119, 2001.

[Cootes 02] T. F. Cootes, G. V. Wheeler, K.N. Walker & C.J. Taylor. *View-based active appearance models*. Image and Vision Computing, vol. 20, no. 9-10, pages 657–664, 2002.

[Cootes 05] T.F. Cootes, C.J. Twining, V.Petrovic, R.Schestowitz & C.J. Taylor. *Groupwise Construction of Appearance Models using Piece-wise Affine Deformations*. In Proc. British Machine Vision Conference, pages 879–888, 2005.

[Cordea 01] Marius D. Cordea, Emil M. Petriu, Nicolas D. Georganas, Dorina C. Petriu & Thomas E. Whalen. *Real-Time 2(1/2)-D Head Pose Recovery for Model-Based Video-Coding*. In IEEE Instrum. and Measure. Tech. Conf, 2001.

[Cristinacce 04] D. Cristinacce & TF Cootes. *A comparison of shape constrained facial feature detectors*. Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, pages 375–380, 2004.

[Cristinacce 06] D. Cristinacce & T. Cootes. *Feature Detection and Tracking with Constrained Local Models*. In 17th British Machine Vision Conference, pages 929–938, 2006.

[Cudalbu 05] C. Cudalbu, B. Anastasiu, R. Radu, R. Cruceanu, E. Schmidt & E. Barth. *Driver monitoring with a single high-speed camera and IR illumination*. In Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on, volume 1, pages 219–222, July 2005.

[Davison 07] A. J. Davison, I. D. Reid, N. D. Molton & O. Stasse. *MonoSLAM: Real-Time Single Camera SLAM.* IEEE Trans. Pattern Anal. Machine Intell., vol. 29, no. 6, 2007.

[Dementhon 95] Daniel F. Dementhon & Larry S. Davis. *Model-based object pose in 25 lines of code.* Int. J. Comput. Vision, vol. 15, no. 1-2, pages 123–141, 1995.

[Dingus 06] T. A. Dingus, S.G. 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.R. Knipling. *The 100-Car Naturalistic Driving Study.* Rapport technique, Virginia Tech Transportation Institute, NHTSA, April 2006. NHTSA NPO-113.

[Dornaika 04] F. Dornaika & F. Davoine. *Head and Facial Animation Tracking using Appearance-Adaptive Models and Particle Filters.* In Computer Vision and Pattern Recognition Workshop, 2004 Conference on, pages 153–153, 2004.

[Doshi 09] Anup Doshi & Mohan Trivedi. *Investigating the relationships between gaze patterns, dynamic vehicle surround analysis, and driver intentions.* IEEE Intelligent Vehicles Symposium, 2009.

[Dowson 05] R. Dowson N.D.H.; Bowden. *Simultaneous modeling and tracking (SMAT) of feature sets.* IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 2, pages 99–105, June 2005.

[Duchowski 02] Andrew T. Duchowski. *A breadth-first survey of eye-tracking applications.* Behavior Research Methods, Instruments, & Computers, vol. 34, no. 4, pages 455–470, 2002.

[Eberly 03] David Eberly. *Fitting 3D Data with a Cylinder.* [web page] `http://www.geometrictools.com/`, 2003.

[EC 03] European Commission. *Communication From the Commission - European Road Safety Action Programme - Halving the number of road accident victims in the European Union by 2010: A shared responsibility.* online, 2003. `http://tinyurl.com/mtad98`.

[Eren 07] H. Eren, U. Celik & M. Poyraz. *Stereo Vision and Statistical Based Behaviour Prediction of Driver.* In Intelligent Vehicles Symposium, 2007 IEEE, pages 657 –662, June 2007.

[Ersal 10] T. Ersal, H.J.A. Fuller, O. Tsimhoni, J.L. Stein & H.K. Fathy. *Model-Based Analysis and Classification of Driver Distraction Under Secondary Tasks.* Intelligent Transportation Systems, IEEE Transactions on, vol. 11, no. 3, pages 692 –701, September 2010.

[ESM 11] ESM. *ESM Safety and Human Factors Investigation and Training.* [web page] `http://www.esm.es/index.php?lang=en`, 2011.

[Fan 07] Xiao Fan, Bao-Cai Yin & Yan-Feng Sun. *Yawning Detection for Monitoring Driver Fatigue.* In Machine Learning and Cybernetics, 2007 International Conference on, volume 2, pages 664 –668, August 2007.

[Farid 06] Mehdi Farid, Ali El Essaili, Matthias Kopf & Heiner Bub. *Methods to Develop a Driver Observation System used in an Active Safety System.* In 22. VDI/VW International Conference on Active Safety and Driver Assistance Systems, Wolfsburg, Germany, Oct 2006.

[Ferrario 02] V. F. Ferrario, C. Sforza, G. Serrao, G. Grassi & E. Mossi. *Active range of motion of the head and cervical spine: a three-dimensional investigation in healthy young adults.* Journal of Orthopaedic Research, vol. 20, no. 1, pages 122–129, 2002.

[Fischler 81] Martin A. Fischler & Robert C. Bolles. *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.* Comm. of the ACM, vol. 24, no. 6, pages 381–395, 1981.

[Fletcher 05] Luke Fletcher & Alexander Zelinsky. *Driver State Monitoring to Mitigate Distraction.* In The Australasian College of Road Safety (ACRS), Stay Safe Committee NSW Parliament International Conference on Driver Distraction, Sydney, Australia, June 2005.

[Fu 06] Yun Fu & Thomas S. Huang. *Graph Embedded Analysis for Head Pose Estimation.* In Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, FGR '06, pages 3–8, Washington, DC, USA, 2006. IEEE Computer Society.

[Gonzalez 02] R. C. Gonzalez & R. E. Woods. Digital image processing. Prentice Hall, 2002.

[Greef 09] Tjerk Greef, Harmen Lafeber, Herre Oostendorp & Jasper Lindenberg. *Eye Movement as Indicators of Mental Workload to Trigger Adaptive Automation.* In Proceedings of the 5th International Conference on Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience: Held as Part of HCI International 2009, FAC '09, pages 219–228, Berlin, Heidelberg, 2009. Springer-Verlag.

[Grest 09] Daniel Grest, Thomas Petersen & Volker Krüger. *A Comparison of Iterative 2D-3D Pose Estimation Methods for Real-Time Applications.* In Image Analysis, volume 5575 of *Lecture Notes in Computer Science*, pages 706–715. Springer Berlin / Heidelberg, 2009.

[Gui 06] Zhenghui Gui & Chao Zhang. *3D Head Pose Estimation Using Non-rigid Structure-from-motion and Point Correspondence.* In TENCON 2006. 2006 IEEE Region 10 Conference, pages 1 –3, November 2006.

[Hansen 10] Dan W. Hansen & Qiang Ji. *In the Eye of the Beholder: A Survey of Models for Eyes and Gaze.* IEEE Trans. PAMI, vol. 32, no. 3, pages 478 –500, mar 2010.

[Harbluk 02] J L Harbluk, Y I Noy & M Eizenman. *The impact of cognitive distraction on driver visual behaviour and vehicle control.* Rapport technique, Transport Canada, february 2002. TP# 13889 E.

[Harbluk 07] Joanne L. Harbluk, Y. Ian Noy, Patricia L. Trbovich & Moshe Eizenman. *An on-road assessment of cognitive distraction: Impacts on drivers' visual behavior and braking performance.* Accident Analysis & Prevention, vol. 39, no. 2, pages 372 – 379, 2007.

[Harris 88] C. Harris & M. Stephens. *A combined corner and edge detector.* In Proc. Fourth Alvey Vision Conference, pages 147–151, 1988.

[Hayhoe 04] Mary M. Hayhoe. *Advances in Relating Eye Movements and Cognition.* Infancy, vol. 6, no. 2, pages 267–274, 2004.

[Henderson 98] J.M. Henderson & A. Hollingworth. *Eye movements during scene viewing: An overview.* Eye guidance in reading and scene perception, vol. 11, pages 269–294, 1998.

[Henderson 03] John M. Henderson. *Human gaze control during real-world scene perception.* Trends in Cognitive Sciences, vol. 7, no. 11, pages 498 – 504, 2003.

[Hoffman 95] J.E. Hoffman & B. Subramaniam. *The role of visual attention in saccadic eye movements.* Perception and Psychophysics, vol. 57, no. 6, pages 787–795, 1995.

[Huang 04] Kohsia S. Huang & Mohan M. Trivedi. *Robust Real-Time Detection, Tracking, and Pose Estimation of Faces in Video Streams.* Pattern Recognition, International Conference on, vol. 3, pages 965–968, 2004.

[Huang 07] He Huang, You-Sheng Zhou, Fan Zhang & Feng-Chen Liu. *An optimized eye locating and tracking system for driver fatigue monitoring.* In Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on, volume 3, pages 1144 –1149, november 2007.

[Itoh 09] M. Itoh. *Individual differences in effects of secondary cognitive activity during driving on temperature at the nose tip.* In Mechatronics and Automation, 2009. ICMA 2009. International Conference on, pages 7–11, August 2009.

[Jalba 04] A.C. Jalba, M.H.F. Wilkinson & J.B.T.M. Roerdink. *Morphological hat-transform scale spaces and their use in pattern classification.* Pattern Recognition, vol. 37, no. 5, pages 901–916, 2004.

[Jepson 03] A.D. Jepson, D.J. Fleet & T.F. El-Maraghi. *Robust Online Appearance Models for Visual Tracking.* IEEE Trans. PAMI, pages 1296–1311, 2003.

[Ji 02] Qiang Ji & Xiaojie Yang. *Real-time eye, gaze, and face pose tracking for monitoring driver vigilance.* Real-Time Imaging, vol. 8, pages 357–377, October 2002.

[Jiao 07] Feng Jiao & Guiming He. *Real-Time Eye Detection and Tracking under Various Light Conditions.* Data Science Journal, vol. 6, pages S636–S640, 2007.

[Jiménez 09] Pedro Jiménez, Jesús Nuevo & Luis M. Bergasa. *Face Tracking and Pose Estimation with Automatic 3D Model Construction.* IET Computer Vision, 2009.

[Kay 98] B. Kay. *How can the Instructor Improve the Long-term Education Process on the Simulator*. In Proceedings INSLC10, pages 3.1–3.4. Institute of Marine Studies, University of Plymouth, 1998.

[Koons 03] D. Koons & M. Flicker. [web page] `http://almaden.ibm.com/cs/blueeyes`, 2003.

[Krinidis 09] M. Krinidis, N. Nikolaidis & I. Pitas. *3-D Head Pose Estimation in Monocular Video Sequences Using Deformable Surfaces and Radial Basis Functions*. IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 2, pages 261 –272, february 2009.

[La Cascia 00] M. La Cascia, S. Sclaroff & V. Athitsos. *Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models*. IEEE Trans. Pattern Anal. Machine Intell., vol. 22, no. 4, pages 322–336, 2000.

[Lander 10] S.A. Lander Simulation & Training Solutions. *TUTOR*. [web page] `http://www.landersimulation.com/eng/solutions/automotive/`, 2010.

[Lanitis 97] A. Lanitis, C. J. Taylor & T. F. Cootes. *Automatic interpretation and coding of face images using flexible models*. IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pages 742–752, 1997.

[Lee 07a] Haet Bit Lee, Jong Min Choi, Jung Soo Kim, Yun Seong Kim, Hyun Jae Baek, Myung Suk Ryu, Ryang Hee Sohn & Kwang Suk Park. *Nonintrusive biosignal measurement system in a vehicle*. Conf Proc IEEE Eng Med Biol Soc, pages 2303–2306, 2007.

[Lee 07b] J. Lee, M. Reyes, Y. Liang & Y.-C. Lee. *Safety vehicles using adaptive interface technology (task 5): Algorithms to assess cognitive distraction*. Rapport technique, University of Iowa, 2007.

[Li 02] Y. Li, S. Gong, J. Sherrah & H. Liddell. *Multi-view face detection using support vector machines and eigenspace modelling*. In Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on, volume 1, pages 241–244. IEEE, 2002.

[Li 04] Yongmin Li, Shaogang Gong, Jamie Sherrah & Heather Liddell. *Support vector machine based multi-view face detection and recognition*. Image and Vision Computing, vol. 22, no. 5, pages 413 – 427, 2004.

[Liang 07] Yulan Liang, M.L. Reyes & J.D. Lee. *Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines*. Intelligent Transportation Systems, IEEE Transactions on, vol. 8, no. 2, pages 340 –350, June 2007.

[Liang 10] Yulan Liang & John D. Lee. *Combining cognitive and visual distraction: Less than the sum of its parts*. Accident Analysis & Prevention, vol. 42, no. 3, pages 881 – 890, 2010. Assessing Safety with Driving Simulators.

[Lin 09] Yi-Tzu Lin, Cheng-Ming Huang, Yi-Ru Chen & Li-Chen Fu. *Real-time face tracking and pose estimation with partitioned sampling and relevance vector machine*. In Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, pages 453 –458, May 2009.

[Lin 10] Yuping Lin, G. Medioni & Jongmoo Choi. *Accurate 3D face reconstruction from weakly calibrated wide baseline images with profile contours*. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1490 –1497, June 2010.

[Lindeberg 98] Tony Lindeberg. *Feature Detection with Automatic Scale Selection*. International Journal of Computer Vision, vol. 30, pages 79–116, 1998.

[Liu 10] Jianping Liu, Chong Zhang & Chongxun Zheng. *EEG-based estimation of mental fatigue by using KPCA-HMM and complexity parameters*. Biomedical Signal Processing and Control, vol. 5, no. 2, pages 124–130, 2010.

[Lourakis 04] M.I.A. Lourakis. *levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++*. [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, 2004.

[Lourakis 09] M. I. A. Lourakis & A. A. Argyros. *SBA: A software package for generic sparse bundle adjustment.* ACM Transactions on Mathematical Software, vol. 1, no. 36, pages 1–30, 2009. Available from `http://www.ics.forth.gr/~lourakis/sba`.

[Lowe 99] David G. Lowe. *Object recognition from local scale-invariant features*. In Intl. Conf. on Computer Vision (ICCV), pages 1150–1157, Corfu, Greece, 1999.

[Lucas 81] B. Lucas & T. Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision*. In Intl. Joint Conf. on AI (IJCAI), pages 674–679, 1981.

[Mahieu 09] Yves Mahieu. *Highlights of the Panorama of Transport*. Rapport technique, Eurostats, 2009.

[Markkula 05] G. Markkula, M. Kutila & J. Engström. *Online Detection of Driver Distraction - Preliminary Results from the AIDE Project*. International Truck & Bus Safety & Security Symposium, November 2005.

[Marquardt 63] Donald W. Marquardt. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. SIAM Journal on Applied Mathematics, vol. 11, no. 2, pages 431–441, 1963.

[Mercedes-B. 08] Mercedes-Benz. *Mercedes-Benz To Introduce Attention Assist Into Series Production In Spring 2009*, August 2008.

[Milborrow 08] S. Milborrow & F. Nicolls. *Locating Facial Features with an Extended Active Shape Model*. In Eur. Conf. on Computer Vision (ECCV), 2008. `http://www.milbo.users.sonic.net/stasm`.

[Miyaji 09] M. Miyaji, H. Kawanaka & K. Oguri. *Driver's cognitive distraction detection using physiological features by the adaboost.* In Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on, pages 1 –6, October 2009.

[Moreels 07] Pierre Moreels & Pietro Perona. *Evaluation of features detectors and descriptors based on 3D objects.* Intl. J. of Computer Vision, vol. 73, no. 3, pages 263–284, 2007.

[Morency 03] L.P. Morency, P. Sundberg & T. Darrell. *Pose estimation using 3D view-based eigenspaces.* In Analysis and Modeling of Faces and Gestures, AMFG 2003. IEEE International Workshop on, pages 45–52, 2003.

[Mostefa 07] D. Mostefa, N. Moreau, K. Choukri, G. Potamianos, S.M. Chu, A. Tyagi, J.R. Casas, J. Turmo, L. Cristoforetti, F. Tobia*et al. The chil audiovisual corpus for lecture and meeting analysis inside smart rooms.* Language Resources and Evaluation, vol. 41, no. 3, pages 389–407, 2007.

[Mourant 72] R.R. Mourant & T.H. Rockwell. *Strategies of visual search by novice and experienced drivers.* Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 14, no. 4, pages 325–335, 1972.

[Mozos 07] Ó. Mozos, A. Gil, M. Ballesta & O. Reinoso. *Interest point detectors for visual SLAM.* Current Topics in Artificial Intelligence, pages 170–179, 2007.

[Murphy-Ch. 07] E. Murphy-Chutorian, A. Doshi & M.M. Trivedi. *Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation.* Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, pages 709–714, 2007.

[Murphy-Ch. 08] E. Murphy-Chutorian & M.M. Trivedi. *HyHOPE: Hybrid Head Orientation and Position Estimation for vision-based driver head tracking.* In Intelligent Vehicles Symposium, 2008 IEEE, pages 512 –517, june 2008.

[Murphy-Ch. 09] E. Murphy-Chutorian & M.M. Trivedi. *Head Pose Estimation in Computer Vision: A Survey.* IEEE Trans. Pattern Anal. Machine Intell., vol. 31, no. 4, pages 607–626, april 2009.

[Neubeck 06] A. Neubeck & L. Van Gool. *Efficient Non-Maximum Suppression.* In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, volume 3, pages 850 –855, 2006.

[Niyogi 96] S. Niyogi & W.T. Freeman. *Example-based head tracking.* In Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG'96), page 374. IEEE Computer Society Washington, DC, USA, 1996.

[Nuevo 09] Jesús Nuevo. *Face Tracking with Active Models for a Driver Monitoring Application.* PhD thesis, University of Alcalá, Department of Electronics, Octubre 2009.

[Nuevo 10] Jesús Nuevo, Luis M. Bergasa & Pedro Jiménez. *RSMAT: Robust simultaneous modeling and tracking*. Pattern Recognition Letters, vol. 31, pages 2455–2463, December 2010.

[Ohayon 06] S. Ohayon & E. Rivlin. *Robust 3D Head Tracking Using Camera Pose Estimation*. In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, volume 1, pages 1063 –1066, 2006.

[Oka 05] K. Oka, Y. Sato, Y. Nakanishi & H. Koike. *Head Pose Estimation System Based on Particle Filtering with Adaptive Diffusion Control*. IEICE Transactions on Information and Systems, pages 1601–1613, 2005.

[Paladini 10] Marco Paladini, Adrien Bartoli & Lourdes Agapito. *Sequential Non-Rigid Structure-from-Motion with the 3D-Implicit Low-Rank Shape Model*. In Kostas Daniilidis, Petros Maragos & Nikos Paragios, editeurs, ECCV, volume 6312 of *Lecture Notes in Computer Science*, pages 15–28. Springer Berlin / Heidelberg, 2010.

[Ranney 01] Thomas A. Ranney, E Mazzai, R Garrott & M J Goodman. *NHTSA Driver Distraction Research: Past, Present, and Future*. Rapport technique, NHTSA, 2001.

[Ranney 08] Thomas A. Ranney. *Driver distraction: a review of the current state-of-knowledge*. Rapport technique, U.S. Dept. of Transportation, National Highway Traffic Safety Administration, Washington, D.C., 2008.

[Rantanen 99] E.M. Rantanen & J.H. Goldberg. *The effect of mental workload on the visual field size and shape*. Ergonomics, vol. 42, no. 6, pages 816–34, 1999.

[Rogers 02] M. Rogers & J. Graham. *Robust active shape model search*. Lecture Notes in Computer Science, pages 517–530, 2002.

[Rongben 04] Wang Rongben, Guo Lie, Tong Bingliang & Jin Lisheng. *Monitoring mouth movement for driver fatigue or distraction with one camera*. In Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on, pages 314 – 319, October 2004.

[Rowley 98] H.A. Rowley, S. Baluja & T. Kanade. *Rotation invariant neural network-based face detection*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 38–44, 1998.

[SafetyNet 08] SafetyNet. Annual statistical report. European Road Safety Observatory, 2008. `www.erso.eu`.

[Sathyanara. 08] A. Sathyanarayana, S. Nageswaren, H. Ghasemzadeh, R. Jafari & J.H.L. Hansen. *Body sensor networks for driver distraction identification*. In Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on, pages 120 –125, September 2008.

[Sathyanara. 10] Amardeep Sathyanarayana, Pinar Boyraz & John H.L. Hansen. *Information fusion for robust context and driver aware active vehicle safety systems*. Information Fusion, vol. In Press, Corrected Proof, pages –, 2010.

[SeeingMach. 10] Seeing Machines. *faceLAB 5*. http://www.seeingmachines.com, August 2010.

[Seemann 04] E. Seemann, K. Nickel & R. Stiefelhagen. *Head pose estimation using stereo vision for human-robot interaction*. In Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, pages 626 – 631, May 2004.

[Seo 04] K. Seo. *Face pose estimation system by combining hybrid ICA-SVM learning and re-registration*. In Asian Conference on Computer Vision, 2004, 2004.

[Sheerman-C. 09] T. Sheerman-Chase, Eng-Jon Ong & R. Bowden. *Online learning of robust facial feature trackers*. In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pages 1386 –1392, October 2009.

[Simard 99] Patrice Y. Simard, Leon Bottou, Patrick Haffner & Yann LeCun. *Boxlets: A fast convolution algorithm for signal processing and neural networks*, 1999.

[Skinner 07] B.T. Skinner, H.T. Nguyen & D.K. Liu. *Classification of EEG Signals Using a Genetic-Based Machine Learning Classifier*. In Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, pages 3120 –3123, August 2007.

[SmartEye 09] SmartEyeAG. *AntiSleep*, 2009. `www.smarteye.se`.

[Stegmann 05] M.B. Stegmann & D. Pedersen. *Bi-temporal 3 D active appearance models with applications to unsupervised ejection fraction estimation*. Proc. SPIE, vol. 5747, pages 336–350, 2005.

[Stutts 01] J C Stutts, D W Reinfurt, L Staplin & E A Rodgman. *The role of driver distraction in traffic crashes*. Rapport technique, AAA Foundation for Traffic Safety, 2001.

[Su 06] Mu-Chun Su, Chao-Yueh Hsiung & De-Yuan Huang. *A Simple Approach to Implementing a System for Monitoring Driver Inattention*. In Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on, volume 1, pages 429 –433, October 2006.

[Sun 07] X. H. Sun, L. Xu & J. Y. Yang. *Driver fatigue alarm based on eye detection and gaze estimation*. MIPPR: Automatic Target Recognition and Image Analysis and Multispectral Image Acquisition, 2007.

[Tango 09] F. Tango, C. Calefato, L. Minin & L. Canovi. *Moving attention from the road: A new methodology for the driver distraction evaluation using machine learning approaches*. In Human System Interactions, 2009. HSI '09. 2nd Conference on, pages 596 –599, May 2009.

[Ting 08] Ping-Huang Ting, Jiun-Ren Hwang, Ji-Liang Doong & Ming-Chang Jeng. *Driver fatigue and highway driving: A simulator study*. Physiology & Behavior, vol. 94, no. 3, pages 448–453, 2008.

[Tomasi 91] Carlo Tomasi & Takeo Kanade. *Detection and Tracking of Point Features*. Rapport technique, Carnegie Mellon University Technical Report CMU-CS-91-132, 1991.

[Torkkola 04] K. Torkkola, N. Massey & C. Wood. *Driver inattention detection through intelligent analysis of readily available sensors*. In Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on, pages 326 – 331, October 2004.

[Tran 10] Cuong Tran & M.M. Trivedi. *Towards a vision-based system exploring 3D driver posture dynamics for driver assistance: Issues and possibilities*. In Intelligent Vehicles Symposium (IV), 2010 IEEE, pages 179 –184, June 2010.

[Triggs 99] B. Triggs, P. McLauchlan, R. Hartley & A. Fitzgibbon. *Bundle Adjustment – A Modern Synthesis*. In W. Triggs, A. Zisserman & R. Szeliski, editeurs, Vision Algorithms: Theory and Practice, LNCS, pages 298–375. Springer Verlag, Sep 1999.

[Triggs 04] B. Triggs. *Detecting keypoints with stable position, orientation, and scale under illumination changes*. Computer Vision-ECCV 2004, pages 100–113, 2004.

[Tu 06] Jilin Tu, T. Huang & Hai Tao. *Accurate Head Pose Tracking in Low Resolution Video*. In Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on, pages 573 –578, April 2006.

[UN-ECE 07] UN-ECE. Statistics of road traffic accidents in europe and north america, volume LI. United Nations, Economic Commission For Europe, 2007.

[Victor 05] T.W. Victor, J.L. Harbluk & J.A. Engström. *Sensitivity of eye-movement measures to in-vehicle task difficulty*. Transportation Research Part F: Traffic Psychology and Behaviour, vol. 8, no. 2, pages 167–190, 2005.

[Viola 04] Paul Viola & Michael J. Jones. *Robust Real-Time Face Detection*. Intl. J. of Computer Vision, vol. 57, no. 2, pages 137–154, 2004.

[Volvo 10] Volvo Car Corporation. *Inside the new Volvo V60 - extra flexibility combined with exclusive quality*, July 2010.

[Vural 07] Esra Vural, Mujdat Cetin, Aytul Ercil, Gwen Littlewort, Marian Bartlett & Javier Movellan. *Drowsy driver detection through facial movement analysis*. In Proceedings of the 2007 IEEE international conference on Human-computer interaction, HCI'07, pages 6–18, Berlin, Heidelberg, 2007. Springer-Verlag.

[Wakita 05] T. Wakita, K. Ozawa, C. Miyajima, K. Igarashi, K. Itou, K. Takeda & F. Itakura. *Driver identification using driving behavior signals*. In Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE, pages 396 – 401, September 2005.

[Wang 07] Jian-Gang Wang & Eric Sung. *EM enhancement of 3D head pose estimated by point at infinity*. Image and Vision Computing, vol. 25, no. 12, pages 1864 – 1874, 2007. The age of human computer interaction.

[Weller 09] G. Weller & B. Schlag. *A robust method to detect driver distraction*. European conference on Human Centred Design for Intelligent Transport Systems, 2009.

[Wesley 10] Avinash Wesley, Dvijesh Shastri & Ioannis Pavlidis. *A novel method to monitor driver's distractions*. In Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems, CHI EA '10, pages 4273–4278, New York, NY, USA, 2010. ACM.

[Wu 00] Ying Wu & K. Toyama. *Wide-range, person- and illumination-insensitive head orientation estimation*. In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pages 183 –188, 2000.

[Wu 08] Junwen Wu & Mohan M. Trivedi. *A two-stage head pose estimation framework and evaluation*. Pattern Recogn., vol. 41, pages 1138–1158, March 2008.

[Xiao 02] Jing Xiao, T. Kanade & J.F. Cohn. *Robust full-motion recovery of head by dynamic templates and re-registration techniques*. In Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, pages 156 –162, May 2002.

[Xiao 04] J. Xiao, S. Baker, I. Matthews & T. Kanade. *Real-time combined 2D+3D active appearance models*. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 535–542, 2004.

[Xiong 05] Yingen Xiong & Francis Quek. *Meeting room configuration and multiple camera calibration in meeting analysis*. In Proceedings of the 7th international conference on Multimodal interfaces, ICMI '05, pages 37–44. ACM, 2005.

[Yang 02a] M.H. Yang, D.J. Kriegman & N. Ahuja. *Detecting faces in images: a survey*. IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 1, pages 34–58, 2002.

[Yang 02b] Ruigang Yang & Zhengyou Zhang. *Model-Based Head Pose Tracking With Stereovision*. In Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, FGR '02, pages 255–, Washington, DC, USA, 2002. IEEE Computer Society.

[Yin 07] Z. Yin & R. Collins. *On-the-fly Object Modeling while Tracking*. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8, 2007.

[Young 07] K. Young & M. Regan. Driver distraction: A review of the literature, pages 379–405. NSW: Australian College of Road Safety, 2007.

[Zhang. 07] Z. Zhang., Y. Hu, M. Liu & T. Huang. *Head pose estimation in seminar room using multi view face detectors.* In Multimodal Technologies for Perception of Humans, Int. Workshop Classification of Events Activities and Relationships, CLEAR, volume 4122, 2007.

[Zhang 08] H. Zhang, M. Smith & R. Dufour. *A final report of safety vehicles using adaptive interface technology (phase ii: Task 7c): Visual distraction.* Delphi Electronics and Safety, 2008.

[Zhao 07] Gangqiang Zhao, Ling Chen, Jie Song & Gencai Chen. *Large head movement tracking using sift-based registration.* In Proceedings of the 15th international conference on Multimedia, MULTIMEDIA '07, pages 807–810, New York, NY, USA, 2007. ACM.

[Zhou 08] Huiping Zhou, M. Itoh & T. Inagaki. *Influence of cognitively distracting activity on driver's eye movement during preparation of changing lanes.* In SICE Annual Conference, 2008, pages 866 –871, August 2008.

[Zitova 03] B. Zitova & J. Flusser. *Image registration methods: a survey.* Image and vision computing, vol. 21, no. 11, pages 977–1000, 2003.