UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Electrónica

Face Tracking with Active Models for a Driver Monitoring Application

A Thesis submitted for the degree of Doctor of Philosophy

Author

Jesús Nuevo Chiquero

Supervisor

Dr. D. Luis Miguel Bergasa Pascual

2009

A mi abuela Pureza

Agradecimientos

Mi primer agradecimiento, como no, va dirigido a Luis Miguel Bergasa Pascual. Un doctorando aprende lo importante que es un buen director de tesis según se avanza en este camino, y yo he tenido mucha suerte. A Luismi tengo que agradecerle su dedicación, especialmente la de estos últimos 2 meses locos, los (pocos) *palos* que me han caído, que a veces vienen bien, y también su esfuerzo y colaboración en conseguir los *premios* de la tesis: las estancias y los congresos a los que he asistido.

El buen ambiente de trabajo y compañerismo que hay en el laboratorio lo crean los miembros del RobeSafe, con quienes tantos buenos ratos he compartido estos años (Pablicio, Llorca, Balky, Gavi, y todos los demás). La grabación de los vídeos utilizados en esta tesis habría sido imposible sin Pedro, Iván y Noelia, y sin la colaboración de los conductores (Vanessa, Pablicio, Dani Pizarro, etc.). Gracias también a Ariadna y Noelia de FicoMirrors S.A. por su colaboración en la captura de datos. Y una mención especial para Sebas por compartir conmigo el *tostón* de marcar los vídeos. Espero que las tecnologías de marcado automático avancen mucho y lo antes posible...

Sin distracciones ni deporte un doctorando se vuelve medio loco. Durante estos años he mejorado bastante poco mi *drive*, pero he pasado grandes ratos jugando al tenis con Nacho, espero que los sigamos jugando en el futuro. Como la música es fundamental, ahí van mis agradecimientos a todos los músicos que me han hecho más llevaderos estos años, especialmente Sufjan Stevens, Radiohead, Ali Farka Touré y Toumani Diabaté.

Viajar es un placer, e irse de estancia más todavía. Gracias a Cedric Pradalier (antes en CSIRO, ahora en ETH Zürich) y a Jonas Sjöberg de Chalmers por darme la oportunidad de visitar sus centros de investigación y acogerme tan bien como lo hicieron. Aprendí mucho y la experiencia estará conmigo siempre.

Muchas gracias a todos mis colegas que me han animado a terminar la tesis (*¿¿¡Cuándo vas a acabar la dichosa*[1] *tesis!??*), especialmente a Iván, Esther y Leyre. Sois increíbles. Me acuerdo también del recién casado PedroTrev, de Iñaki, Miguel do Brasil, Mike y de Amelio, que es la viva demostración de que con pasión y esfuerzo se puede llegar allá donde te propongas.

Y por último, que no por ello menos, quiero dar las gracias a mis padres y a mi hermana. Creo que no tenían muy claro dónde me metía cuando empecé el doctorado y siguen sin saber a dónde lleva esto (yo tampoco lo sé), pero que aún así me apoyaron siempre y me dieron su cariño sin dudar.

Los profesores son mis héroes
Teachers are my heroes

Alea jacta non est

[1] entre otros adjetivos

Resumen

La falta de atención durante la conducción es una de las principales causas de accidentes de tráfico. La monitorización del conductor para detectar inatención es un problema complejo, que incluye elementos fisiológicos y de comportamiento. Se han realizado diferentes aproximaciones a este problema y entre ellas la Visión Computacional ofrece la posibilidad de monitorizar a la persona al volante sin interferir con su conducción. Una estimación precisa del estado del conductor se puede obtener analizando expresiones faciales, movimientos oculares y acciones como el parpadeo y la fijación de la mirada.

Un sistema de visión computacional para la monitorización del conductor utiliza en sus primeras etapas de procesado la detección de la cara y el seguimiento de su posición con el tiempo. El empleo de técnicas adecuadas en estas fases es de vital importancia para el buen funcionamiento del resto del sistema. Esta tesis presenta un método para el seguimiento de la cara, utilizando modelos activos para caracterizar la faz del usuario. Tres tipos de modelos se han analizado sobre un nuevo conjunto de vídeos de monitorización del conductor llamado RS-DMV, propuesto en esta tesis. La base de vídeos se compone de secuencias grabadas en un vehículo real, y en 2 simuladores realistas. En los primeros vídeos, los sujetos realizaron acciones habituales de la conducción. Los conductores en el primer simulador se vieron expuestos a situaciones que destacaban comportamientos de distracción. Finalmente, los conductores en el segundo simulador se encontraban en estado de privación de sueño, y mostraban claros signos de fatiga y somnolencia.

El objetivo de RS-DMV es crear una base de pruebas que contenga situaciones típicas de la aplicación objetivo, para comparar el rendimiendo de los modelos y métodos analizados y el cumplimiento de los requerimientos y restricciones de un sistema en producción: ejecución en *tiempo real*, robustez frente a oclusiones y giros de cabeza, expresiones faciales, una iluminación cambiante y unos usuarios de diferente apariencia.

Se ha analizado el rendimiento de los Modelos Activos de Forma (ASM) y de los Modelos Locales Restringidos (CLM), por considerarlos a priori de interés. En concreto, se ha evaluado el método *Stacked Trimmed* ASM (STASM), que integra un número de extensiones sobre la propuesta original de ASM, mostrando una alta precisión en todos las pruebas cuando la cara es frontal. Sus principales problemas son que no funciona cuando la cara está girada, y que los tiempos de proceso están lejos de los requerimientos de *tiempo real*. CLM se ejecuta en *tiempo real*, pero tiene menor precisión que STASM, y tiene que ser reposicionado de manera regular. Tampoco es capaz de ajustarse adecuadamente durante giros de cabeza.

El tercer método a evaluar es el Modelado y Seguimiento Simultáneo (SMAT), que caracteriza la forma y la textura de manera incremental, a partir de muestras encontradas previamente. La textura alrededor de cada punto de la forma se modela mediante un conjunto de grupos (*clusters*) de muestras pasadas. La forma de la cara se describe también con su propio conjunto de *clusters*. Este modelado facial dinámico se adapta bien a los requerimientos de la aplicación, y ha sido tomado como base para nuestra

propuesta. A partir del mismo se han realizado diversas aportaciones en esta tesis. Se han desarrollado 3 métodos de *clustering* alternativos para modelar la textura, que en los tests obtienen errores de ajuste menores que el original, y son más fiables. Se ha integrado dentro de SMAT un modelo de forma obtenido a partir de un conjunto de muestras de entrenamiento, como el empleado en los ASM. Se ha aplicado un estimador-M para un ajuste robusto de la forma, lográndose una mejora ostensible en la robustez del modelo frente a giros de cabeza y oclusiones. Este nuevo método lo hemos denominado SMAT Robusto (R-SMAT).

Se ha evaludado la mejora del método R-SMAT con respecto al SMAT original y se ha comparado el rendimiento de R-SMAT y STASM sobre los vídeos de RS-DMV. En conclusión, R-SMAT es capaz de procesar más de 100 imágenes por segundo, y obtiene una precisión similar a STASM y muy superior al SMAT original.

Abstract

Driver inattention is one of the main causes of traffic accidents. Monitoring a driver to detect inattention is a complex problem that involves physiological and behavioural elements. Different approaches have been made, and among them Computer Vision has the potential of monitoring the person behind the wheel without interfering with her driving. An accurate estimation of the state of the driver can be obtained by analyzing facial expressions, eye movements and actions like blinking and gaze fixation.

A computer vision system for driving monitoring uses face location and tracking as the first processing stages. Relaying on adequate techniques in these stages is key for the correct operation of the rest of the system. This thesis presents a method for face tracking, using active models to characterize a face. Three different models are tested on the newly composed RobeSafe Driver Monitoring Video (RS-DMV) dataset, proposed in this thesis. The dataset contains sequences of drivers in real scenarios and two realistic simulators. The first sequences present subjects performing the most common actions in everyday driving. Drivers on the first simulator were presented with situations that highlighted distracted behaviours. Finally, the drivers in the second simulator had been deprived of sleep, and were in fatigue and showed signs of drowsiness.

The aim of RS-DMV is to create a test set containing situations that appear frequently an application of driver monitoring. This set can be used to compare the performance of models and methods, and the fulfillment of the requirements and restrictions of a production system: real-time execution, robustness to occlusions and head turns, facial expressions and changing illumination, and users of diverse appearance.

The performance of Active Shape Models (ASM) and Constrained Local Models (CLM) have been tested, as they have been considered of interest a priori. More precisely, the Stacked Trimmed ASM (STASM), which integrates a number of extensions to the original ASM, has demonstrated great accuracy in all tests when the face is frontal. Its main drawbacks is that it does not work when the face is rotated, and processing times are far from the required real-time execution. CLM runs in real-time, but it is less accurate than STASM, and needs periodic repositioning. It is not able to handle head turns properly.

The third method evaluated is Simultaneous Modeling and Tracking (SMAT), which characterizes shape and texture incrementally from samples previously encountered. The texture around each point of the shape is modeled with a set of clusters of previous samples. The shape is also described with its own set of clusters. This dynamic face model is well suited for the requirements of the application, and has been used as the base for our proposal. Several contributions have been made on this method. Three alternative clustering methods have been developed to model texture, and in the test they obtain lower fitting error, with better reliability. A shape model built from a training set as the one used by ASM has been integrated in SMAT. An M-estimator has been included for robust shape fitting, and a remarkable improvement in the overall robustness of the model to head turns and occlusions has been achieved. We have denominated this new method

Robust SMAT (R-SMAT).

The improvements of R-SMAT over the original SMAT have been evaluated, and the performance of R-SMAT and STASM has been compared on the sequences in RS-DMV. In conclusion, R-SMAT is able to process more than 100 frames per second, obtains similar accuracy to STASM, and much better than the original SMAT.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Automotive industries have a long standing tradition of innovative research and development. Ever increasing competition and investment has led to many advances in various fields of technology and science, from aerodynamics to engines, from materials to control. With the gains made in later decades is reliability and performance, manufacturers have focused even more on safety, and joined institutions and governments in a common goal of reducing the number of traffic accidents and the severity of these.

Traffic safety is indeed an enormous problem for society. 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]. In Spain alone, 2,741 people died in 2007, and 2,181 in 2008 in traffic accidents. In an effort to reduce these figures, the European Commission set up in 2003 the European Road Safety Action Programme (2003-2010) [European Commision 03], which aims to halve the number of victims in road accidents by 2010.

Inattention is a major cause of traffic accidents, and groups distraction and sleepiness. Distraction can be divided in two main types: visual and cognitive. Visual distraction is straightforward, occurring when drivers look away from the roadway (e.g., to configure a GPS device or sound system). Cognitive distraction occurs when drivers think about something not directly related to the current vehicle control task (e.g., conversing on a hands-free cell phone or route planning). Cognitive distraction impairs the ability of drivers to detect targets across the entire visual scene and causes gaze to be concentrated in the center of the driving scene. Table 1.1 shows the increased risk of accident of various inattentive behaviours and actions. Inattention has been found to be involved in some form in 80 percent of the crashes and 65 percent of the near crashes within 3 seconds of the event [Dingus 06]. Missouri Department of Transportation [Missouri DoT 07] also identifies inattention as the most frequent cause involved in accidents, based on crash reports. According to the U.S. National Highway Traffic Safety Administration (NHTSA), at least 100,000 automobile crashes annually are caused in the U.S. by falling asleep while driving. An annual average of roughly 70,000 nonfatal injuries and 1,550 fatalities results from these crashes [Royal 03, Rau 05]. These figures only cover crashes happening between midnight and 6 am, involving a single vehicle and a sober driver traveling alone. This time frame is also the one with lower traffic density, and other studies [Flatley 04] have found a correlation between low traffic density in motorways and an increased number of sleep-related accidents.

Behaviour	Crash or near-crash risk increase
Drowsiness	4x
Reaching a moving object	9x
Looking at external object	3.7x
Reading	3x
Dialing a hand-held device	3x
Talking on a hand-held device	1.3x

Table 1.1: Increased risk of inattentive behaviours. *(Source: Virginia Tech Transp. Institute-NHTSA [Dingus 06]).*

Crashes involving distraction are likely to rise in the near future, due to the increasing use of in-vehicle information devices, such as GPS navigation systems, satellite radios and DVD players. These devices are additional sources of distraction, and enabling drivers to benefit from them without diminishing safety is an important challenge in their design.

Intelligent systems able to detect driver sleepiness, distraction or both are needed to reduce the number of traffic crashes. It is a complex problem. Estimating human actions and cognitive states is difficult and changes from person to person, and depends on factors such as age, physiological state or even cultural background. Research groups and car manufacturers alike have been working on different approaches to the problem in the last years. Approaches and problems are presented in the next section.

1.2 Inattention detection issues and approaches

Many researchers have worked in recent years on systems for driver inattention detection, focused mainly in drowsiness, with a broad range of techniques. Sleep has a long history of research in the fields of psychology and medicine, where accurate measurements and indicators have been developed [Rechtschaffen 98, U. of California 97]. Electroencephalograms (EEG) [Klein 07, Susmáková 04] represent the electrical changes in the brain, measured with a series of electrodes placed in the scalp. The electrodes detect small voltages produced in the brain cortex. These potentials form waves at several frequencies, known as delta, theta, alpha, beta and gamma waves, that are linked to different cognitive and motor processes, including drowsiness and the different sleep stages, as shown in figure 1.1. Brain studies couple EEG with electrooculography (EOG), which detects eye movements, and electromyogram (EMG) that monitors muscular tone.

These measurements provide the best data for detection of drowsiness, and as such have been used by several drowsiness detection systems, usually in conjunction with heart rate and breathing rate. The problem of these techniques is that they are intrusive to the subject. They require electrodes and other sensors to be placed on the head, face and chest as in figure 1.2, which may annoy the driver. They also need to be carefully placed: installing the electrodes to obtain an EEG requires external help and takes a few minutes, and medical equipment is always expensive. Recent research has introduced some contact-less readings, but no remarkable results have been achieved so far. Nonetheless, physiological measures such as EEG have been used in some projects [Kircher 02], and are frequently used as the ground-truth for testing other, less invasive methods.

A driver's state of attention can also be characterized using indirect measurements and contact-less sensors. Lateral position of the vehicle inside the lane, steering wheel movements and time-to-line crossing are commonly used, and some commercial systems

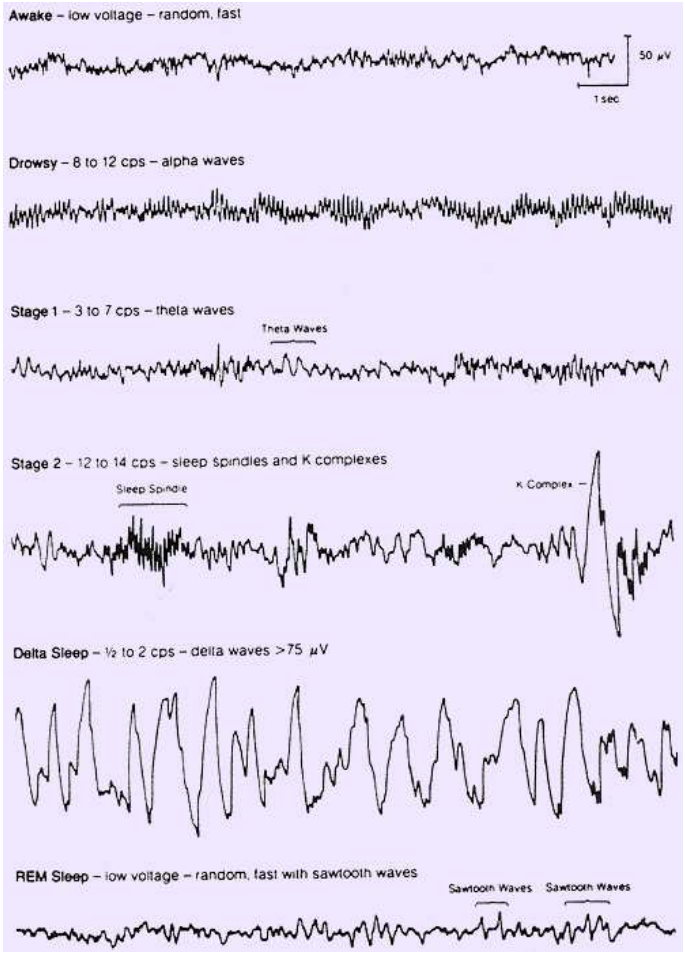


Figure 1.1: An example of brain wave activity for different sleep stages. *(Source: www.sleephomepages.org)*

have been developed. These systems do not monitor the driver's condition, but its driving. Volvo Cars introduced its Driver Alert Control system [Volvo Car Corp. 08] in 2008, which is available on its high-end models. This system uses a camera, a number of sensors and a central unit to monitor the movements of the car within the road lane, to assesses whether the driver is drowsy (see figure 1.3). Mercedes-Benz has introduced a similar system (ATTENTION ASSIST) [DaimerAG 09] in its newest E-Class vehicles (shown in figure 1.4). Daimler AG, owner of Mercedes-Benz, was in 2001 one of the first to develop a system based on vehicle speed, steering angle and vehicle position relative to road delimitation (recorded by a camera) to detect if the vehicle is about to leave the road [DaimerAG 01]. Other manufactures have conducted research and presented prototypes. Toyota [Kircher 02] used steering wheel movement sensors and pulse sensor to record the heart rate. Mitsubishi has reported the use of steering wheel sensors and measures of vehicle behavior (such as lateral position of the car) to detect driver drowsiness in their advanced safety vehicle system [Kircher 02].

These techniques are not invasive, and to date they obtain the most reliable results with the least number of false positives, a critical problem in this type of systems. However, they face several limitations such as geometric characteristics and state of the road, and driver experience. They also require a training period for each person, during which the driving style of the user is learned and modeled, and thus are not applicable to the

Figure 1.2: A patient wearing a helmet with electrodes for EEG.

occasional driver. Despite the high number of parameters involved, these systems are basic in that the behaviours they are able to detect are few: the measurements may not reflect user behaviours such as the so-called micro-sleeps: if a drowsy driver falls asleep for few seconds, the lateral position may not change in a straight road [Ueno 94]. Response time of these systems may compromise their effectiveness.

Drivers in fatigue exhibit changes in the way their eyes perform some actions, like moving or blinking. These actions are known as *visual behaviors*, and are readily observable in drowsy drivers, and also in distracted ones. More precisely, typical characteristics include longer blink duration, modified blinking frequency, slow eyelid movement, a smaller degree of eye opening and gaze (narrowed field of view, with reduced response to objects in the peripheral areas of vision). Although not purely *visual*, other characteristics that are included in this group are yawning, nodding, sluggish facial expression (due to relaxed muscular tone) and dropping posture. Of all of them, the percent of eye closure (PERCLOS) has been found to be the most reliable indicator of drowsiness [Dinges 98].

Computer vision has been the tool of choice for many researchers to be used to monitor visual behaviours, as is non-intrusive. Most systems use one or two cameras to track the head and eyes of the subject [Matsumoto 00, Victor 01, Kutila 06]. Commercial products are available for general applications not focused on driving problems. A few companies commercialize systems as accessories for installation in vehicles, but are not part of the car manufacturers' developments: reliability is not high enough for car companies to take on the responsibility of its production and possible liability in case of malfunctioning. By installing the system themselves, the owners take the responsibility instead. Seeing Machines sells the FaceLAB software [Seeing Machines 04] that uses two cameras to track the face in 3D. They have also presented the Driver State Sensor (DSS) [Seeing Machines 07],

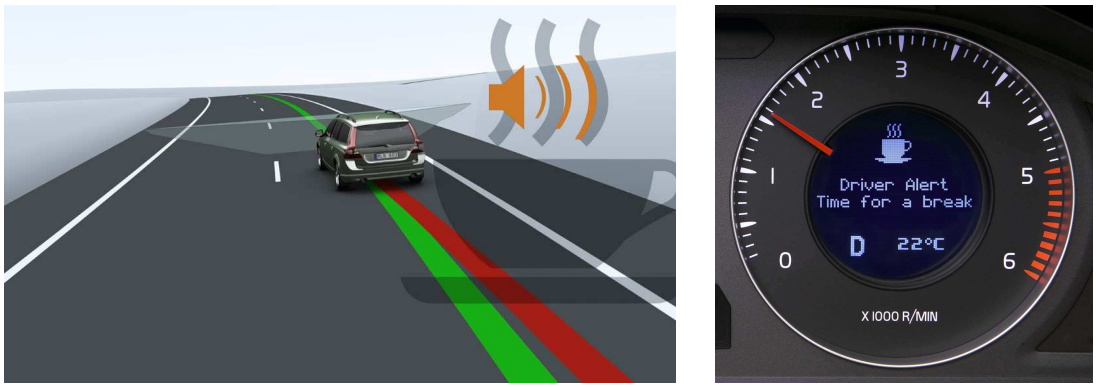
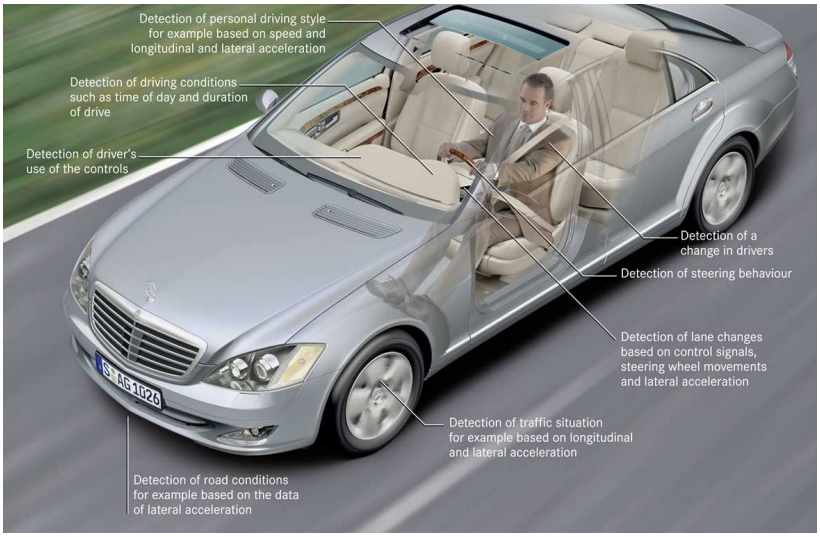


Figure 1.3: Volvo's Driver Alert Control. *Images from Volvo Cars*



(a)

Figure 1.4: Mercedes-Benz's Attention Assist. *Images from emercedesbenz.com*

which calculates PERCLOS. The Swedish company SmartEye AG [SmartEyeAG 09] offers mono- and multi-camera systems that detect eye movements, gaze fixation and blink detection. Mono-camera systems have been a major focus on late years, because integration in industrial production is much easier and less costly.

As it can be seen, some systems have indeed entered the market, but in the literature there are very few details available regarding the methods and parameters of those systems.

Computer vision systems use natural light, infra-red (IR) or both to illuminate the face of the driver. This is an important problem of system that must work 24/7 and day and night scenarios are very different. Usually daytime algorithms need to be adapted to work during nighttime. [Shih 00] presented a system using 3D techniques to estimate and track the line of sight of a person using multiple cameras. In [Ji 02] a system with active IR illumination and a camera is implemented. In addition to providing illumination, IR light reflects on the eye's cornea and produces the *red-eye effect*, similar to the one appearing in photography when flash light is used. This reflection can be detected and used for locating and tracking the rest of the face. They propose to estimate the local gaze direction based

on pupil location. In [D'Orazio 07] a system based on natural light was presented.

Systems relying on a single visual cue may fail when the required features can not be detected accurately or reliably. Also, people's visual behaviours under fatigue or distraction change from person to person, and a single indicator may not be representative of the overall cognitive state [Ji 02]. Relaying on multiple visual cues reduces the uncertainty and the ambiguity compared to that of relaying on only one source. Recent research points in this direction. In [Boyraz 08], another multi-sensor system was presented. The authors tested two decision making methods, fuzzy inference system (FIS) and artificial neural networks (ANN) to fuse the data and obtain an estimation of the drowsiness state of the driver. The AWAKE European project [AWAKE Consortium 04] proposed a multi-sensor system that integrated multiple visual cues with information from the vehicle and the environment. This system must be configured explicitly for each driver, requiring a learning stage. Another European initiative, SENSATION [SENSATION 07], carried on with this line of research. A non-intrusive system fusing driver's condition information with data from his/her driving, with minimal to no per-person customization would be the best candidate for mainstream adoption, and thus research has concentrated on this option lately.

1.3 Objectives of this thesis

From 2004, the RobeSafe Research Group[1] at the Department of Electronics, has been working in the problem of estimating the alertness level of drivers [Bergasa 04]. Several important results were achieved, and an automatic active-illumination system based on one camera that used up to 7 parameters was developed [Bergasa 06]. The most significant indicators were found to be PERCLOS and fixed gaze. The focus of that system was to detect drowsiness, although the addition of the fixed gaze visual cue would make it possible to detect some kinds of cognitive distractions. Drivers simulated drowsiness in real driving situations, and detection rate was over 90%. The eye location and tracking was based on the *red-eye effect* on the drivers' pupil, that was produced with the help of near-IR LED illumination. The system was able to work reliably at night, and on drivers without glasses, as glasses reflect the IR light and make the eye much more difficult to detect. During the day, sunlight intensity was much higher than that of the IR reflection, eyes could not be detected and the system was not able to work properly.

The face and eye detection step is at the base of the image processing of the system. To make it work during daytime and with drivers wearing glasses, good face tracking and eye detection methods are required.

Human face detection and tracking is indeed a broad field in computing research [Yang 02], and a myriad of techniques have been developed in the last decades. It is of the greatest importance and interest, as vast amounts of information are contained in face features, movements and gestures, which are constantly used for human communication. Face detection and tracking is the first step for other algorithms that work in face recognition or expression analysis in some areas of computer vision [Belhumeur 97, Buenaposada 08].

To date, most of the published techniques for face tracking have been demonstrated in indoor environments, under more or less controlled illumination and with limited movements. Only in recent years methods tested outdoors have appeared, with some of them using images or video sequences that feature drivers in a moving vehicle. From this situation, two closely related problems can be considered. First, the necessity of developing

[1] www.robesafe.com

detection and tracking techniques able to work properly on drivers' faces in a real driving situation, and second, developing a common test ground, in the form of a video sequence database representative of the problem under study, so the performance of different methods can be properly compared.

The aim of this thesis is to develop a computer vision method able to detect and track the face of a driver in a robust fashion, minimizing the number of tracking losses, and with the highest precision possible. It is to serve as the bases of an automatic driver inattention detection system, which will work together with the system presented in [Bergasa 06]. The tracker must continue working when head turns, partial occlusions and illumination changes take place, in both day and night scenarios.

1.3.1 System requirements

The fact that the final objective of this thesis is to develop a system that can be integrated in a production vehicle imposes a series of restrictions on the system. Of the multiple elements and options that can be part of the system, a configuration has been chosen that uses *off-the-shelf* components for reduced costs. Options with possible implantation problems, such as stereo systems (that require calibration), or infra-red cameras have been discarded.

The requisites of the system to be developed are the following:

- **One camera**, of visible spectrum with reasonable sensitivity to near-IR, able to provide at least 30 frames per second.
- **Real-time** operation, at the frame rate of the camera.
- **Automatic operation with any user**.
- **Daytime and nighttime** operation.
- **Robust operation** in presence of **head turns** and movements.
- **Robust operation** in presence of **partial occlusions**, either by the driver's hands or any other external element.
- **Robust operation** in presence of **illumination changes**.
- **Automatic and fast localization of the face** after tracking losses.

1.4 Document structure

This document is divided in several parts, of which the present introduction is the first one. Chapter 2 contains a brief review of the state of the art in face detection and tracking. That chapter aims to present a global view of the approaches to face tracking, without going into specific details. These are introduced as needed in the following chapters, as related techniques and methods are described.

Chapter 3 introduces the RobeSafe Driver Monitoring Video (RS-DMV) dataset, made of recordings of different people as they drive an utility vehicle around the University Campus, and in some realistic driving simulators. The database characteristics are described, as well as error evaluation methods and comparisons.

Chapter 4 presents the characteristics of the Active Shape Model (ASM) and Constrained Local Models(CLM), and tests their performance on the sequences of the RS-DMV dataset. Chapter 5 presents the Simultaneous Modeling and Tracking (SMAT) method, and our proposal based on the latter. Newly developed modifications over this algorithm are explained, and their performance compared over the video sequence database.

Finally, chapter 6 contains the conclusions and main contributions of this work, and future research lines that may spring from it. An appendix and bibliography close this document.

Chapter 2

State of the Art

Face detection in static or moving images, and its tracking in the later case, have been some of the most active fields of research in computer vision for the last 3 decades. As it was mentioned in the introduction above, the information that can be obtained from a human face is extreme valuable for many applications, from computer security or access control identification to psychology studies. The number of methods that have been developed is huge. Moreover, human face is arguably the most common example of deformable object, and as such it has been used to demonstrate the performance of many generic methods that deal with deformable surfaces or objects.

This chapter presents a brief survey of the state of the art in face detection, and face tracking in video sequences. In relation to face tracking, it also reviews some of the most successful techniques in face modeling. This chapter does not intend to make an exhaustive review, as it would result in a lengthy chapter, both in time and space. Several authors have published extended surveys of the literature, which will be referenced below, where the reader could find more detailed information on some works, and additional references to past publications not mentioned here. The aim of what follows is to provide an overview of the most remarkable methods in each field, and those that are related to the contents of the following sections of this thesis. On those sections, related methods to the presented proposal will be explained in more detail. This chapter closes with a discussion on the most adequate methods to be studied for the intended application of this work, and the specific aims of this thesis.

2.1 Face detection

Face detection deals with the problem of finding if human faces appear in an image, and locating them if any are present. Almost every system that extracts and analyzes the information contained in the face has face detection as its first step. Advances in later years have made face detection evolve into a technology, and has extended visibly in many consumer orientated products: nowadays, virtually every digital camera, laptop and camera-phone sports this functionality.

Given the diversity of the methods published in the literature, different classifications of them are possible. Classification of some of them is not always evident, and there is a certain degree of overlapping between groups. We separate some of the most interesting approaches in the following categories, loosely based in the survey of [Yang 02]. We also refer to the survey of [Hjelmas 01] for alternative subdivisions, and [Yang 08] for more references.

- **Feature-based methods:** Characteristic features present in the human face are searched for in the image, and a face is detected based on a set of rules involving those features. Features used are usually chosen so they are independent of point of view, illumination or expression.
- **Template-based methods:** These methods try to locate the face by comparing parts of the image to patterns learned in advance.
- **Classifier-based methods:** These methods train a discriminant classifier in advance, such as an Artificial Neural Network [Rowley 96], and process the input image with it.

2.1.1 Facial feature-based methods

An extensive number of methods have been presented that use facial features to localize the face. This bottom-up procedure tries to find components of the face, such as the eyes, eyebrows or nose, and then applies a set of pre-learned rules to determine the presence of a face. These rules are usually expressed as distance and position relationships, modeled statistically as mutual distances [Burl 95] or as a shape [Kendall 84].

A large amount of research has used low level features such as edges and blobs [Chetverikov 92], or local detectors. In [Burl 95], eyes, nostrils and mouth are detected with facial templates defined by the pixel response to multi-scale and multi-orientation Gaussian filters. [Yow 97] presented a method where features are found using a derivative Gaussian filter. Edges are searched for in the surroundings of the feature, and then a Bayesian network was used to evaluate the validity of the groups of each feature and grouping.

Skin color has been used by several methods as the basis for face detection. One of the main problems of using color is the influence of illumination, which has been tried to be reduced in various ways. In [McKenna 98], skin tones are modeled with a mixture of Gaussians in the HS color space. Other works have used other spaces, such as normalized RGB [Crowley 97, Kim 98], RG [Martinkauppi 02, Bergasa 00] or YCbCr [Chai 99, Tsapatsoulis 00], to reduce the effect of illumination in the skin color. The Gaussians mixture model is trained using the EM algorithm [McLachlan 97] with pixel values extracted from sample images. [Bergasa 00] presented an unsupervised and adaptive Gaussian skin color model, based on a simplification of the EM algorithm. In [Jones 02], the authors built RGB histograms using thousands of pictures from the Web, and used it to detect skin pixels on images. Fuzzy sets were used to detect faces in color images in [Wu 99], by identifying skin and hair. More recently, [Tsalakanidou 05] presented a system that combined skin color detection with depth data obtained with a 3D range sensor, for a more precise localization of the face.

2.1.2 Template-based methods

Human face shows a series of structured components that remain relatively unaltered, such as its oval shape or contours corresponding to eyes and mouth. Template-based techniques search for these elements using predefined patterns, extracted or built from training sets of images. In [Craw 87], a method using shape templates of a frontal face was presented. The template was applied on the edge image obtained with the Sobel operator. More complex techniques, such as silhouettes [Samal 95] or active contours (*snakes*) [Gunn 94, Lam 94] have also been applied to the problem.

[Lanitis 95] presented a face modeling method based on shape and intensity data. Landmark points in the face represented in a point distribution model (PDM), learned from a training set of manually marked shapes using Principal Component Analysis (PCA). The shape is used to search faces and estimate its shape parameters using an Active Shape Model (ASM) [Cootes 95]. ASM has been later extended by other researchers to include robust fitting functions [Rogers 02] and stacked models [Milborrow 08].

2.1.3 Classifier-based methods

Machine learning research has greatly developed in late years, and its advances have been applied to face detection. Some algorithms in this group have shown impressive results. The best known is the Adaboost-based technique of Viola & Jones [Viola 04]. Their method uses Haar-like features computed efficiently over the whole image. Using a cascade of weak classifiers removes the most improbable instances early, reducing the computational load and enabling for fast detection. This algorithm shows good performance with partially occluded faces. Extensions to this method have been presented [Pham 07a], partially solving the problem of its lengthy training process [Viola 02, Pham 07b]. As a matter of fact, Viola & Jones' algorithm and other methods that developed its framework [Huang 07] have been so successful that research in the field of face detection has greatly reduced, and the amount of publications reported in the literature has fallen sharply in the last 5 years. An implementation and trained classifiers for faces are available in the OpenCV library [Bradski 08].

Other methods using classifiers have also been presented. Neural networks were used in [Burel 94, Rowley 96] with good results, and have also seen extended use. Support Vector Machines have been applied [Romdhani 01, Osuna 97] too, as well as Naive Bayes classifiers [Schneiderman 04].

2.2 Face tracking

Application of tracking techniques to human faces in video sequences has several objectives, from the simplest reduction in processing time by constraining the search area in continuous localization, to more complex of pose estimation or extracting movement trajectories that may define human behaviours of interest.

Standard filtering techniques, such as Kalman or Particle filters have been used by many authors [McKenna 95, Strom 99, Zhou 02]. It makes sense to consider in the classification of the techniques the kind of features or data that they use as reference. The groups below are partially inspired by those in the survey of [Wang 03], to which we also refer for more references.

2.2.1 Facial features-based methods

Tracking facial features has been used by several authors as a mean to track the face as a whole. Works involving features are common in research aimed at human-machine interfaces. Eyes (or parts of them), being a distinctive feature that best expresses attention, have been the focus of several studies. [Ji 02] presented a system that tracked the eyes with a Kalman filter using the reflection of near-IR light on the pupil of a person (the *red-eye* effect) as the measurement, a technique also used in [Bergasa 06], and by some commercial products [Tobii 04]. A particle filter was used in an active contour tracker in [Hansen 05], and a dual-state model that tracked the eyes whether they are open or closed

was presented in [Tian 00]. In addition to eyes, eyebrows [Gee 94], nose [Gorodnichy 02] and mouth have also been used.

2.2.2 Skin color-based methods

As was the case in face detection, skin color has been extensively used in face tracking methods. Modeling of skin color is still done with Gaussian models [McKenna 99] or histograms in various color spaces [Birchfield 98, Perez 02]. [Buenaposada 01] developed a dynamic grey-world algorithm [Buchsbaum 80] to reduce the influence of illumination as the face moved in a video sequence. An adaptive extension to the Mean Shift algorithm was presented in [Bradski 98] and used in a human-machine interface to track the face, with the objective of having a low computational load. [Bergasa 00] uses a linear combination of previous Gaussian skin model parameters to predict their new values, and a zero-order Kalman filter over the shape face to track the face in the image. [Yang 96] approach includes three models, of color changes due to illumination, face movement and camera to increase the robustness of the tracking.

2.2.3 Template-based methods

Tracking the face using the most salient parts of it has a greater chance of success than relaying on other, less defined parts. One of the most common approaches is to use templates of these elements. [Jebara 97] tracked the nose, and mouth and eye corners using correlation and Kalman Filters. Structure-from-motion was used to generate a 3D model, constrained by parametrized models trained in advance.

General purpose template-based trackers have been used to track faces in the literature with success. Tracking with templates can be a computationally intensive task, and great efforts have been devoted to reduce the processing load of the algorithms. Solving the correspondence between template and image is usually posed as minimizing a Sum of Squared Differences (SSD). [Hager 98] approached the problem by factorizing the Jacobian into a constant part and a non-constant part that is cheaper to compute. This optimization only works with affine projections, [Buenaposada 04] presented an alternative factorization of the Jacobian that allows for a projective model to be used. An illumination invariant extension of this method [Buenaposada 06] has been tested on a limited number of video sequences of drivers [Bergasa 08]. Another efficient approach is the *inverse-compositional* [Baker 01] that uses Gauss-Newton minimization and obtains a constant Hessian by composing the update to the parameters as an additional transformation, instead of considering the update as an addition. On the other hand, [Jurie 02] introduced a compositional update method where the Jacobian is numerically computed.

Active Shape Models (ASM) [Cootes 92, Cootes 95] are deformable models that have been demonstrated on faces with good results. These models are similar to the active contours (*snakes*), but include constraints from a Point Distribution Model (PDM) [Dryden 98] computed in advance from a training set. 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 modeling of texture have been presented, of which Active Appearance Models (AAMs) [Cootes 01a] are arguably the best known. These have been used on color images [Stegmann 03b] and edge maps [Cootes 01b], and some experiments tracking faces have been performed [Matthews 04b, Dornaika 04]. Few tests employed images recorded on cars [Baker 04c], but these sequences were short and did not contain challenging scenarios. Both ASM and AAM are linear models, and have dif-

ficulties tracking faces under occlusions, or self-occlusions during head turns. View-based models have been developed to treat these cases separately [Cootes 02, Morency 03]. Applying M-estimators [Huber 81], robust versions of these models able to work under partial occlusions have been presented [Gross 04]. Usually Principal Component Analysis (PCA) is used to build the model, although other dimensionality-reducing techniques have been applied [Uzümcü 03].

Active Appearance Models are global models in the sense that the minimization is performed over all pixels that fall inside the mesh defined by the mean of the Point Distribution Model. Triangulated meshes are widely used. In the case of a face, this involves areas that have little texture, or that are far away and thus subject to respond differently to illumination changes or movements.

To solve these problems, Cootes *et al.* proposed the Constrained Local Model (CLM) [Cristinacce 06] that only considers small areas around the landmarks. Patches centered on the landmarks are fitted to the image using normalized cross-correlation, subject to the restrictions imposed by the PDM. CLM obtains better performance than AAMs.

All the methods above have an offline training stage, where the model is built from (most commonly) handmarked data. This stage can be very time consuming if the number of images to be labeled is high. Some works have tried to skip this by building a model online, with the Simultaneous Modeling and Tracking [Dowson 05] being one of the most recently presented. These methods only work on video sequences.

While 2D models are the most common, three-dimensional face models have also been presented [Blanz 99a]. These models usually include range data, obtained with 3D range laser scans. Although most work has been devoted to fitting these models to still images, some application to face tracking has been presented [Paterson 03]. Simpler 3D models, such as the cylindrical one in [La Cascia 00] have also been used for tracking and to obtain the face pose with good results.

2.2.4 Motion-based methods

Optic flow has been used by several authors to track human faces, usually coupled with a facial model. [Li 93] proposed a method to extract rigid and non-rigid motion from a sequence using a 3D mesh placed over the face, once the 3D position of the point had been obtained. [Black 97] used a 2D patch model to track both rigid and not rigid motion, the latest been used later for expression recognition. An ellipsoidal 3D model was presented in [Basu 96].

2.3 Discussion

Previous sections have introduced a number of published methods for face detection and tracking. Even though there are many more, it has been shown that the diversity of the approaches is high. However, there are few works focused on the intended application of this thesis and its specific characteristics, stated in the introduction above.

Face detection research seems to have peaked with the introduction of Viola & Jones method [Viola 04], that exhibits good performance and robustness. The availability of implementations and trained classifiers for faces (among other objects), allows for a very fast application of this solution in any problem that requires face detection. For small images it works in or close to real-time, and their authors acknowledge a performance of 91.4% with 50 false positives on the MIT+CMU image set [Viola 04]. This database contains

507 frontal faces. Another algorithm based on the Viola & Jones framework [Huang 07], reported 97% of success with less than 100 false positives in the CMU profile testing set [Rowley 98], that includes faces in multiple poses. For our purposed application, the face will be frontal to the cameras most of the time and in up-right position, so the original Viola & Jones algorithm can be considered good enough, and used as face detector in our system.

Face tracking is a more open problem than face detection. In order to gain a general view of the performance of these methods, table 2.1 shows a comparison of some of the algorithms referred above, in terms of their characteristics and reported tests and performance, and whether they run in real-time or not. Error values are given by authors in several different forms. Precision is normally presented as the mean error in pixels and/or degrees, while some systems that estimate a discrete number of face poses report a percentage of the poses correctly determined.

Many authors do not provide numeric error values of their algorithm tracking a human face, and only present pictures and/or movies of their systems in action. These works are indicated with *samples* in the error column of the table. Evaluation is performed in most works with images captured indoors. Some authors use freely available image sets, but most of them test on internal datasets created by them, which limits the validity of a comparison with other systems. Only a few authors [Cristinacce 06][Bergasa 08] have used images recorded in a vehicle, but the number of samples is limited. Please refer to the original works for more details.

Some of the methods report processing times for old hardware that is no longer available, and from which it may be difficult to estimate the processing times in current CPUs. For simplicity, only algorithms that are reported to be able to run in real-time (or safely expected to do so) are indicated as such on the table. This *property* of the algorithms is, as can be seen from the table, one of the most widespread, which reflects the focus on developing methods that are usable in real applications.

One characteristic that divides the methods based on facial features and skin color from those using templates or motion is the use of a face model. This way, certain parts of the model would always correspond to an specific part of the face. While this characteristic can be obtained through other means, models that require training offline can have a semantic meaning, as parts of them trained e.g. for the nose, will correspond to the nose on the images if the tracking is successful. Not all models, however, have the same representative power ([La Cascia 00] does not guarantee this hypothesis). Because the monitoring system to be developed will need to recognize the eyes and other parts of the face and extract information from them, this property is of great interest, and thus we propose to use a facial model in this thesis.

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. Few trained models are openly available [Milborrow 08], as there are marked image databases [FGNet 04]. This problem also extends to 3D morphable models, specially those using laser scanners, because these machines are very expensive. Using trained models already available, or models that require minimal training in the development of the face tracking system would be desirable for our purposes.

Robustness is a major focus of most works. Most are tested indoors. Illumination changes in the test sets are performed with one moving light source, casting different shadows on the faces. Occlusion tests involve covering part of the face with a book or a hand, the latter being a bigger challenge as the skin tone is similar to that of the

Approach	Error	Evaluated	Tracking	Robust to	Requirements	Other
Features						
Red-eye effect[Ji 02][Bergasa 06]	$\sim$2%	Indoor([Ji 02]),outdoor([Bergasa 06])	KF	ILL	UC([Ji 02]),AI	RT
Act.Contour[Hansen 05]	$< 5px$	Indoor	PF	OCC+ILL	-	RT
Nose tracking[Gorodnichy 02]	*samples*	Indoor	-	NT	-	RT
Skin color						
Gaussian model[McKenna 99]	*samples*	Indoor	KF	ILL	TO	RT
Gaussian model+EM[Bergasa 00]	$\sim 2\%$	Indoor	KF	ILL	-	RT
Histograms[Birchfield 98]	*samples*	Indoor	CV	OCC	TO	RT
Grey-world[Buenaposada 01]	*samples*	Indoor	-	ILL	TO	RT
CAMSHIFT[Bradski 98]	negligible in position	Indoor	-	OCC	TO	RT
Templates						
Corr.+SfM[Jebara 97]	*samples*	Indoor	EKF	OCC	-	RT,FM
Jacobian Optim.[Hager 98]	*samples*	Indoor	-	OCC+ILL	-	RT
Jacobian Optim. [Buenaposada 04]	*samples*	Indoor+Outdoor	-	OCC+ILL	-	RT
ASM[Milborrow 08]	$< 0.1{m_e}^1$ in $\sim 95\%$ of tests	Indoor	-	OCC+ILL	TO	RT,FM
AAM[Cootes 01a]	$RMS = 0.008{m_e}^{12}$	Indoor	-	-	TO	RT,FM
CLM[Cristinacce 06]	$< 0.1{m_e}^1$ in $\sim 90\%$ of tests	Indoor+Outdoor	-	OCC+ILL	TO	RT,FM
3D cylindrical model[La Cascia 00]	$< 2in, < 5^\circ$	Indoor	-	ILL	-	RT,FM
Motion						
Optic flow 2D[Li 93]	*samples*	Indoor	optic flow	ILL	-	FM
Motion regularization+3D[Basu 96]	*samples*	Indoor	optic flow	OCC	-	FM

AI: Active illumination. **CV**: Constant Velocity, calculated from two last positions. **FM**: face model. **ILL**: (robust to) illumination changes. **NT**: no data, or not tested. **OCC**: (robust to) occlusion. **RT**: Real-time ($\geq 20fps$). **TO**: system or model is trained offline. **UC**: requires user calibration.

[1] $< \mathbf{0.1\ m_e}$ indicates the mean error is below 0.1 the inter-eye distance.

[2] RMS error of converged tests. Convergence criteria was $< 0.05\, m_e$. See paper for more details.

Table 2.1: A comparison of face tracking methods

face. Methods that model the texture holistically (for the whole face) seem to have more difficulties coping with illumination changes, as these are local in most cases. Some works have tried to address this problem [Le Gallou 06] by preprocessing the texture. Still, models that fit locally obtain better results, either with patches (like CLM) or profiles (like ASM). Of the methods listed in table 2.1, only CLM and ASM are based on local fitting, and for this reason these two methods will serve as the basis for our proposal.

As can be seen in the table, most works provide a series of images from video sequences of their systems working. A few of them use openly available databases for some tests, but most of them are just collections of still images, and researchers test their methods on sequences recorded by them in their labs. Most are recorded indoors, and to the best of our knowledge, there is no available database of images or movies recorded in a moving car in an actual road, which is the expected scenario where a system such as ours should work.

2.4 Aim of this thesis

After the review of the state of the art, and considering the requisites presented in the introduction, the aims of this thesis are as follow:

1. To date, there are no available datasets of video sequences recorded in a real-vehicle moving in a real road. A sufficiently representative dataset is to be recorded, with different drivers in a real scenario. The database will be used to test the methods proposed in this work, and then made available to the public.

2. To research the adequacy of techniques like ASM and CLM, which have been considered of interest a priori, for the intended application of this thesis, i.e. tracking the face of a driver in a real scenario.

3. Model training is time consuming and it may also exhibit problems when the user does not resemble the appearances covered by the images in the training set. It is an aim of this thesis to research models that require simple training, or no a priori training at all, with similar performance to methods that have an offline training stage.

4. To develop a automatic face detection and tracking system with minimal training that will serve as the basis for an application of visual driver monitoring. It must comply with the restrictions of a production system (work with any user, real-time execution, robustness to occlusions, head turns, illumination changes, night and day operation and failure detection and recovery) which are not dealt with in most of the system in the state of the art.

5. To assess the performance of the proposed system using the new dataset and exhaustive evaluation. The system is to be compared with other techniques presented in the literature.

Chapter 3

Performance evaluation and Video Data Set

Evaluating the performance of a computer vision algorithm is a complex task [Clark 04], and different approaches have been made since the first works in Computer Vision. These are intimately related to the ever-increasing computing capabilities of modern processors and systems. In the beginnings of the Computer Vision research in the 1960s, available computing resources were so little that algorithms where severely limited in the amount of data they could work with, and most performance evaluations were purely analytical. A decade or so later, researchers were already able to perform much more complex operations on images, but still no extensive comparisons between algorithms could be done.

The vast processing resources of today's computers have made possible to evaluate the performance of a method quantitatively. This form of evaluation is by far the most common in Computer Vision research, as there are usually too many unknown variables and assumptions on the images, which make the qualitative evaluation infeasible for all but the simplest algorithms, with a few exceptions. *Performance characterization* is the process of discovering how the characteristics of the data sets used by the algorithm affect it, and to what extent.

One of the main problems in assessing the performance of a computer vision algorithm is then the choice of the data that will make the *test set*, and also the *training set* in the case that the algorithm has an initial learning step. Many recent works use one or more of the databases available to the public, although it is also very common to introduce a new data set in every new piece of research. This is sometimes justified when the available databases do not cover the requirements of the new developments, but it can also be an obstacle in obtaining a clear comparison between methods.

This thesis evaluates different methods for face tracking, and compares their performance. As these methods will be executed on a common set of images and videos, this chapter introduces the tests that will be performed, the error measurements, and how the methods will be compared. Then, it presents the details of the data sets used, and why they have been chosen. This chapter closes with conclusions and contributions.

3.1 Performance evaluation

Methods for face location and tracking are evaluated based on the distance between the estimated face location and the true one. The threshold to be set in that case is the error value below which the algorithm is considered to have succeeded, or failed if the error is

above that limit.

3.1.1 Performance evaluation of face tracking methods

Methods used for face tracking are presented in chapters 4 and 5. Without getting into the details of these methods, all of them represent the face with a model that includes the position of some characteristic points, also known as *landmarks*. The elements of this *point distribution* are not predefined, but are usually very similar if not the same, and are placed on salient features of the face: eye and mouth corners, nose and eyebrows, among others, as can be seen in figure 3.1. (See chapter 4.1 for a more precise definition of *landmarks*)

A quick review of the literature [Cootes 01a, Cristinacce 04], [Matthews 04b], and also [Stegmann 03a, Stegmann 05], [Dowson 05] shows that all authors consider the performance of their algorithms as a function of the distance between the estimated position of the landmarks, and their actual position, the *ground-truth*. Ground-truth values are usually hand marked by a human operator.

Figure 3.1: An example of point placement

There are, however, differences in how the error values are calculated, and how the convergence of the algorithms to a correct solution is defined. In [Cootes 01a], Cootes *et al.* use the *Root Mean Square* of the distance between points and their corresponding ground-truth values as error measurement, usually noted as *RMS-PE*. They consider that the algorithm converges if the RMS-PE is below 5% of the width of the face they are trying to locate and model. They report that the average size of the frontal face in their images is around 200 pixels. Matthews and Baker [Baker 04a] use the same error measurement, RMS-PE, but define convergence when the error is below 1.0 pixels.

Stegmann *et al.* in [Stegmann 03a] use the unsigned mean of two different distances: between the points and their ground-truth, and between the points and the curve that links the ground-truth values. In [Stegmann 05] they develop an extended 3D model, and use the error in the estimated volume as the error measurement. Other works that use the mean of the point-to-point Euclidean distance as the error measurement include [Cristinacce 04, Cristinacce 06]. In these works by Cristinacce *et al.* they introduce a

scaling factor in the measurements:

$$m_e = \frac{1}{ns}\sum_{i=1}^{n} d_i, \qquad d_i = \sqrt{(\mathbf{x}_i - \hat{\mathbf{x}}_i)^T(\mathbf{x}_i - \hat{\mathbf{x}}_i)} \tag{3.1}$$

where $\mathbf{x}_i$ are the coordinates of point i, $\hat{\mathbf{x}}$ the position estimated by the algorithm, d_i is the error for point i, n is the number of points and s is a scaling factor, that 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, as this distance is not affected by any deformation of the face due to gestures. This scaling factor compensates for the apparent variation in size when the person is closer or further away from the camera, but may not be accurate when the face rotates around the vertical axis. In that case, the distance between the eyes reduces, but the size of the face in the image does not. Face is frontal in all the images they use in their work, so no additional error is introduced by the scaling factor.

Considering the previous examples, the error measurement m_e in equation 3.1 could be the most appropriate for its consideration of the scale. However, as will be shown later, the image and video data sets used in this thesis try to replicate the behavior of a driver in a real situation as close as possible, and that includes pronounced head turns. Still, m_e is used in this work as the basic error measurement for the face tracking methods: when the head appears turned in an image, the inter-eye distance is estimated by hand from previous values.

Whether a precise definition of convergence is given in any particular work in the literature, most authors present a result curve of the percentage of test that succeeded against the convergence threshold. While not properly a *Receiver Operating Characteristic (ROC)* curve, this graph gives a clear idea of the amount of error in the position estimation that can be expected for the required success rate. An example is shown in figure 3.2.

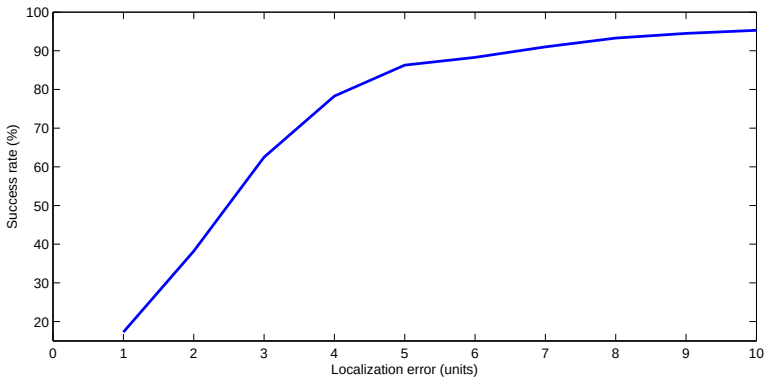


Figure 3.2: Curve relating the success rate and the localization error

Performance comparisons are done in this thesis by superposing the result curves obtained for the different algorithms and configurations. There may be cases where curves will cross, as is the case in figure 3.3, and then the election of the most appropriate algorithm may no be straight forward. In these situations, there is a compromise between the maximum tolerable error level and the increase in the success rate, that depends on the subsequent processing stages where the results of the algorithms will be used, and thus each particular case will be discussed on its own.

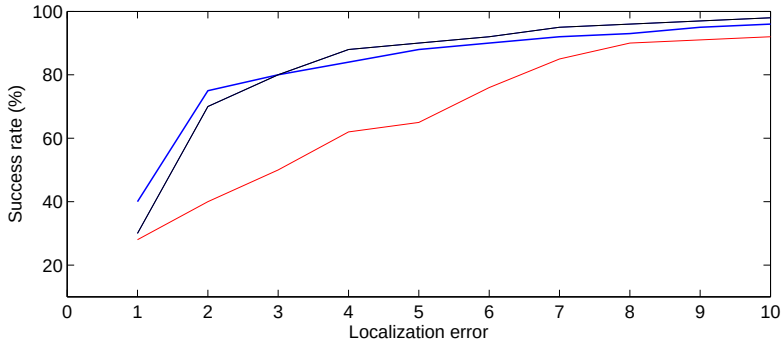


Figure 3.3: Comparing algorithms by their result curves

It should be noted that point distributions used by the methods studied in this work differ in size. The Active Shape Model implementation (Stacked Trimmed ASM, STASM [Milborrow 08]) discussed at the beginning of chapter 4 uses 68 points, while Constrained Local Models (also in chapter 4) use only 20. Simultaneous Modeling and Tracking (SMAT), in chapter 5 use the same 20 points as CLM. Cristinacce and Cootes in [Cristinacce 06] discard 3 of these points from the error evaluation, because their positions varies greatly from person to person, and their positions when given by human markers have great variance. This rule has been followed also for SMAT. As for ASM, the STASM implementation includes an option of producing an error estimation based on the 17 points use in CLM[1]. These points are shown in figure 3.4, with discarded points in red. The error measurement is noted as m_e17. When a method is processing an image of a partially occluded face, points that are not visible are not considered for the error computation.

In addition to evaluating the error with the m_e17 measurement, the frequency of tracking losses for the different algorithms is also calculated. The evolution of the error and detected tracking losses are provided for some sequences with images of interesting moments, so situations where the performance decreases can be easily identified.

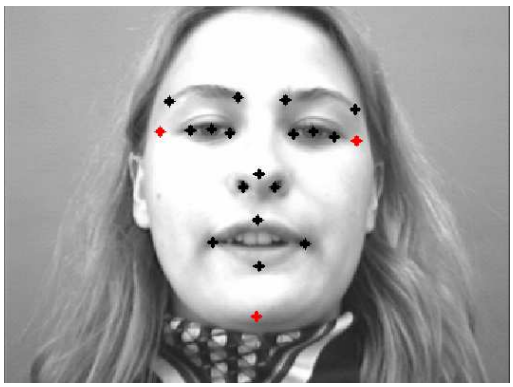


Figure 3.4: CLM point distribution

Algorithms tested in this thesis require and initial, rough estimation of the location of the face and its features. This estimation can be obtained either for a human operator or another fitting algorithm. In our case, the STASM initializes with Viola & Jones and is

[1]The STASM source code was modified to produce results in such a way that they are compatible with the rest of the data from CLM and SMAT.

able to work automatically. It has been subsequently used to initialize CLM and SMAT.

3.2 Databases and ground truth values

The choice of the data that will be used to test an algorithm is critical in Computer Vision, because the behavior of the methods in this field of research is highly data dependent. Because of the complexity of the human face, a quality database must be of sufficient size, and be able to represent the variability of different parameters of interest, such as illumination, pose, expression and identity. Collecting the large amounts of data required is a resource intensive task.

The specific characteristics of the algorithm to be tested may determine the choice of one database over another, depending on the problem that the algorithm is targeting. For instance, a face recognition method that is expected to work outdoors under uncontrolled lighting should not be tested with images taken in a laboratory with constant illumination. As such, image databases are usually designed with a focus on a particular topic (face detection, recognition, gesture detection, etc.), or a few of them.

In recent years, an increasing number of publicly available face image databases have been presented [Jesorsky 01, Martinez 99, Blanz 99b, Sim 03, Belhumeur 97]. Some of them include a set of annotated points that can be taken as ground-truth [Jesorsky 01, Martinez 99]. An extensive review of many of them was presented in [Gross 05a]. Almost all of these databases provide 2D images of subjects, plus additional metadata such as the type of illumination, the expression on the face of the person, and the pose of the face. A few provide 3D face scans [Blanz 99b, MISKL 05, FRAV 09]. However, fewer video databases [Kleiner 04, O'Toole 05] are available, in part because still images are enough to cover the necessities for testing of many algorithms that may run on video sequences.

The characteristics of the work presented in this thesis led to an evaluation of some of the available databases, of which a few of the most representative are briefly described below. Images in them exhibit illumination changes, occlusions, self-occlusions and various expressions, on different persons recorded over a span of several days. None of them contain, however, elements that can not be ignored in the intended application of this thesis, that is, monitoring drivers in everyday driving situations. Those elements include motion blur, blur from (de)focus, background changes, people showing fatigue or distraction, and wearing glasses. Day and nighttime operation must also be taken into account.

Motion blur occurs when objects being recorded move fast enough so that its position changes noticeably during the capture time of the camera, i.e., while the sensor is being exposed. The amount of blur depends on the speed of the object, and on the capture time. Because low-light driving environments (nighttime, tunnels, shadows, etc.) are common and some may occur unexpectedly, the shutter time is set over 10ms, where motion effects are visible. It must also be considered that the cameras used have a small sensor size, around 1/4 of an inch, which limits the amount of light they can absorb.

To increase the amount of light that impacts the sensor, the second camera setting that can be adjusted is the aperture. Increasing the aperture has a drawback in that the *depth of field* [Wikipedia 08] is reduced. With the driver located so close to the camera (around 1 meter), the depth of field with a large aperture is very shallow. In some cases, the driver may get closer to the driving wheel, and thus the camera, getting out of focus. The third setting that impacts the pixel levels is the gain of the sensor. Its drawback is that noise increases rapidly with gain, reducing the quality of the images.

To the best of the candidate's knowledge, there is no publicly available video dataset

of people driving, either in a simulator or in a real road. For this reason, a set of videos of different subjects has been recorded in driving simulators and in a moving car. The sequences cover most of the driving scenarios, and are described below. This set will be used in the tests of the algorithms in this document.

Following some details of four of the most common available face databases are presented. They are discussed briefly. It is not the purpose of this section to give an in-depth review of these image sets, but to give a sample of the publicly available databases and the reasoning behind them being used or discarded. We refer to the publications and webpages of the databases' authors for more information. The new set of recorded videos is described in detail in the next section.

3.2.1 AR database

The AR database [Martinez 99], was created at the Computer Vision Center (CVC) of the University of Barcelona in 1998, by Aleix Martinez and Robert Benavente. It contains over 4000 color images of 126 people's faces (70 men and 56 women). The images have a size of 768x576 pixels in 24-bit of depth RGB color, free of compression. Two sessions per person were done, with a time lapse of two weeks between them, with 14 shots taken in each session. The images were captured under carefully controlled conditions. The background is plain.

Each subject was asked to show 4 different expressions, neutral, smile, anger and scream. Then, images were taken with lights on the right and left turned on alternatively, without occlusion of the face, and wearing sun glasses and a scarf. In all cases, the subject was frontal to the camera. An example is shown in figure 3.5.

Figure 3.5: An example of the AR database. (*From the AR website*)

Annotations of some images are available from the Face and Gesture Recognition Working Group[2]. The annotations contain 22 points per image. As of this writing, only images showing expressions 1, 2, 3 and 5 have been marked.

This database has been widely used (over 200 research groups are reported have downloaded it) in face detection and recognition research [Yang 02]. The number of subjects that make part of the database is quite high. However, there are only 4 types of illumination cast on the subjects' faces. This, and the fact that in all images the face is frontal to the camera, reduces the interest of this database as the test set to be used.

3.2.2 CMU-PIE database

The Carnegie Mellon University (CMU) Pose, Illumination and Expression (PIE) database [Sim 03] contains 41,368 color images of 68 people. Each person's image was captured from

[2] `http://www-prima.inrialpes.fr/FGnet/data/05-ARFace/tarfd_markup.html`

13 different poses, 43 different illumination conditions and exhibiting 4 facial expressions, at the CMU 3D Room [Kanade 98] with 13 synchronized cameras. Image size is 640x480, in RGB color.

Figure 3.6: An example of the CMU-PIE database (*From the CMU-PIE website*)

The position of all elements in the room was measured using a theodolite and is provided with the database. Background images were captured to aid processing, and color calibration data was collected to minimize the differences between the cameras. No markup points are available.

While the calibration, the multiple view images and the illumination scheme make this database a very good candidate to work with, the absence of markup data is a fair obstacle for this database to be used in the tests.

3.2.3 Cohn-Kanade AU-Coded Face Expression Database

The Cohn-Kanade Active-Unit-Coded Face Expression Database [Kanade 00] contains images of over 180 subjects of various ethnicity, that performed a series of face expressions. The subjects were told to produce facial expressions that were identified and coded with the Facial Action Coding System (FACS) [Ekman 78]. Frontal-view images were captured with a resolution of 640x480 pixels, and a short video sequence was taken from a 30-degree view. A few samples are shown in figure 3.7. Manual annotations of the images are available from the LAIV group at the Universitá degli Studi di Milano[3].

Figure 3.7: Samples from Cohn-Kanade Database. ©Jeffrey Cohn

As its name indicates, this database is focused on face expressions, and has been widely used in this field of research [Tian 01, Gross 01, Liu 03, Buenaposada 08]. Face

[3] `http://lipori.dsi.unimi.it/download/gt2.html`

expression recognition is outside the scope of this work, and the fact that the images and video sequences show little to no head movements makes this database not adequate for the purposes of this thesis.

3.2.4 BioID database

The BioID database [Jesorsky 01] contains 1521 frontal images of 23 different subjects. The images were recorded in real world situations, with very different backgrounds and illumination conditions. The position of the camera with respect to the subjects varies from top-down view to bottom-up, and the faces are not all at the same distance from the camera. The database do not provides any pose information of the camera and subjects. A few samples can be seen in figure 3.8.

Figure 3.8: An example of the BioID database. (*From the BioID website*)

A 20 point markup of all images in the database was manually made by David Cristinacce and Kola Babalola, and is available from the Face and Gesture Recognition Working Group[4]. The ground truth position of the eyes is also available from the database website.

Although all faces are frontal in this dataset, the different environments where the images were taken, with complex illumination and background, make it a good candidate for it to be used in some of the tests. The availability of the hand-marked points is also of great help in assessing the performance of the algorithms.

3.3 RobeSafe Driver Monitoring Video Dataset

As part of different projects of the RobeSafe Research Group at the Department of Electronics, which the candidate is with, some series of videos of drivers have been recorded. Many of these were captured in environments such as a driving simulator, which imposed particular conditions on the drivers and their mental and physical state, and also on the hardware setup. Of all videos available, 14 have been selected for being representative of different actions and behaviors that take place in real driving. These sequences make the RobeSafe Driver Monitoring Video Dataset (RS-DMV), and are described below.

3.3.1 Driving scenarios

The objectives of this thesis require a diverse set of test data. In one hand, testing face tracking in a car requires a set of videos recorded outdoor, with changing illumination and shadows, quick movements and occlusions, in a car moving on a street or highway. On

[4] `http://www-prima.inrialpes.fr/FGnet/data/11-BioID/bioid_points.html`

the other hand, drivers exhibiting inattentive behaviours can not be captured on video while driving on a street, because it is extremely dangerous, even if it is just *simulated* inattention.

There are several options to overcome this problem, in part at least. One would be to record the videos on a car moving in a closed track. This is, at the moment, unfeasible for cost reasons and still represents a high-risk activity for the driver, even at low speeds. Another would be to capture the video stream for a subject on the co-driver seat, instead of the driver itself. This option has many advantages, however it has been discarded because psychological studies [Lal 02, Wierwille 96, Rau 05] about drowsiness and distraction in drivers focus on the driver herself, and there are important differences on the mental workload for the driver and the co-driver, and its influence can not be underestimated[5]. For example, the motivation for the driver not to fall asleep is obviously much stronger than for the co-driver.

The database is split in three types of videos, depending on the driving scenario and the physical condition of the drivers.

Type A - Outdoors

Type A videos were recorded outdoor, on RobeSafe's vehicle moving at the campus of the University of Alcala. These videos were recorded under the MOVI^{2}CON project[6] (`TRA2005-08529-C02-02`).

Drivers were fully awake, talked frequently with other passengers in the vehicle and were asked to look frequently to the rear-view mirrors and operate the car sound system. These sequences try to capture common actions that take place in everyday driving. All subjects drove the same streets, shown in figure 3.9. The length of the track is around 1.1 km.

Figure 3.9: Streets at the University Campus where the videos were recorded (map from *maps.google.com*)

The weather conditions during the recordings were mostly sunny, which made noticeable shadows appear on the half of the face further away from the window. Global illumination changes took place as the car moved, due to the presence of trees by the road. Local illumination changes affecting only part of the face occurred when the driver's head

[5]While it could be possible to study the differences in mental workload between the driver and the co-driver, and develop ways to reduce their disparity, it is a task that falls much closer to psychology and physiology, and thus is outside the scope of this work.

[6]`http://www.dia.fi.upm.es/~movicon/`

moved closer or further away from the window. Two images of the car on the streets are shown in figure 3.10

Figure 3.10: The car used in the outdoor recordings

A stereo camera system was built and installed in the car, as depicted in figure 3.11. The baseline of the system is 18cm. The cameras are two monochrome Basler 1400-17fm, with an IEEE-1394b (FireWire 800) interface. The focal length of the lenses is 9mm. This focal length allowed for the subject to appear with a convenient size in the images. An external synchronization signal was generated using a micro-controller. Frame size is 960×480, and frame rate is 30 fps.

(a) The stereo system was calibrated (b) A view of the system from outside

Figure 3.11: Stereo camera setup, as installed in the car

Type B - Truck simulator

Type B videos were recorded in a truck simulator located at the Centro de Estudios e Investigaciones Técnicas de Gipuzkoa (CEIT) in San Sebastian, Basque Country, under the CABINTEC project[7] (PSE-370100-2007-2). The simulator is high-fidelity, with set-up composed of a six-axis movement platform, simulation cabin and a visual system. The movement platform is able to withstand loads up to 1,000 kg. The cabin, in figure 3.12, mimics a real truck cabin. The visual system is made of three retro-projection screens

[7] www.cabintec.net

(1 frontal, 2 lateral) and covers the 180º of the driver's field of view. TFT screens are installed outside the cabin as rear-view mirrors, as shown in figure 3.12(d).

(a) Right-hand view (b) Left-hand view

(c) Frontal view (d) Visual system

Figure 3.12: Truck simulator

Drivers were fully awake, and were presented with a demanding driving environment were many other vehicles were present and potentially dangerous situations took place. These situations increase the probability of small periods of distraction leading to crashes or near-crashes. The sequences try to capture both distracted behaviour and the reaction to dangerous driving situations. Additionally, the sequences can be taken as a nighttime scenario because of the low-light conditions. Diffused near-infrared illumination was used to increase the amount of light available to the cameras. The LED-based illuminators were placed close to the cameras, and synchronized with the exposure triggers with a micro-controller.

The camera system is the same to that of the *type A* scenario, with the exception that 12mm lenses are used, because the cabin size put the driver further away from the cameras. Updated camera firmware allowed to increase the frame size to 1392×480 pixels, with the same frame rate of 30 fps.

Type C - Passenger vehicle simulator

Type C videos were recorded in a car simulator. These videos were recorded in collaboration with the international company FICOMIRRORS, S.A.[8] and the Instituto de Biomecánica de Valencia (IBV)[9].

The simulator consisted in a bench with a driver seat, a seat-belt to fasten the subject, and driving simulator software. Figure 3.13 shows two images of the simulator. In addition to the images, a series of physiological data was captured: skin temperature, electro-encephalogram, electro-oculogram, two derivations of electro-cardiogram, pulsoximetry, and muscular activity of the upper body associated to the sympathetic-vagal system.

Figure 3.13: Two images of the car simulator

Drivers were in fatigue, and showed signs of drowsiness. Subjects in the recordings were deprived of sleep, after their workday and had been awake for the 24 previous hours. The same night environment was simulated in all cases, with only an artificial dim light, and a stable temperature around 24-26 °C. A monotonous road sound at low volume was played during the simulation, to further induce drowsiness. Near-IR illumination was used, but with lower intensity than in the truck simulator, and no *bright pupil effect* is present.

These videos were captured using a mono-camera system, with a 12mm lens. Frame size is 1332×720 pixels, and frame rate is again 30 fps.

Compromises in the recording of sequences

In order to widen the possible applications of the recorded sequences in the different projects as much as possible, some compromises had to be made between the different objectives of these projects. A system designed specifically for the tests to be carried

[8] www.ficosa.com
[9] www.ibv.org

out in this document would have had a single camera facing the subject, without any additional cameras or sensors, but other set-ups were used to obtain stereo images and physiological data. Some of the most remarkable deviations from the *ideal* set-up for this thesis are:

- **Stereo camera recording system:** Cameras were positioned for *type A* and *type B* as shown in figures 3.11(b) and 3.12(c), and then no straight frontal view of the face can be obtained from any camera. This camera position may also increase the amount of background visible on the image. In the case of the camera on the right-hand side the window is clearly visible, and the background may change quickly and noticeably.

- **Image size:** The images have been recorded in high resolution, which takes considerable storage space. It also increases the amount of data to be moved from the hard-drive to memory when reading the videos. High resolution is not necessary carry out face tracking, but it is of help when estimating other parameters like the width of the eyes and pupils, or the percentage of closure of the eyes.

- **Infrared (IR) illumination:** Some videos were recorded in very low light conditions, and diffused infrared (IR) illumination was used on the subjects' faces to increase the amount of light available to the cameras. In some cases, the *red-eye* effect appears in the subject's pupils. On subjects wearing glasses, reflections of the IR source also appear in the glasses. Diffused IR illumination is a good solution in low light conditions, because it is not perceived by the subject and then does not interfere with the driving. The reflections on the glasses saturate the values of the image pixels, and occlude the parts of the face behind them, that unfortunately fall in the eye area.

- **Biometric sensors:** Subjects that appear in *type C* videos wear different biometric sensors that are clearly visible on the images. These sensors acquired brain signals, and eye and heart data that were needed for other experiments.

With the exception of the IR light reflection on the glasses, these compromises result in minor or no changes on the images from what would be obtained from the preferred system, and do not pose significant problems. Images are resized by the algorithms as needed.

3.3.2 Videos in the database

The main characteristics of the database are the following:

- The database is composed by 14 videos. Of them, 7 were recorded outdoors (*type A*), 3 in the truck simulator (*type B*) and another 4 in the car simulator (*type C*).

- Videos were recorded with 11 different subjects. Of them, 3 are female and the rest male.

- Video size depends on the length of the track, traffic conditions and the speed of the vehicles. *Type A* videos are around 2 minutes long. *Type B* videos are the longest, around 10 minutes long, and *type C* are around 5 minutes.

- Videos are compressed with a lossless codec (FFV1), that reduces the file size by at least a factor of 2.

Table 3.1 summarizes the details of the videos in the RS-DMV dataset. Samples and details from a few of these sequences are given below.

Figure 3.14 shows some samples from *type A* videos. The subjects talk frequently, and focus their attention on the rear-view mirrors and car radio several times during the recordings. Two subjects wear glasses. Fast illumination changes occur, specially during turns as the relative position of the sun changes quickly.

(a) Sequence #1 (b) Sequence #1 (c) Sequence #2

(d) Sequence #4 (e) Sequence #4 (f) Sequence #5

(g) Sequence #5 (h) Sequence #7 (i) Sequence #7

Figure 3.14: Samples of *type A* videos (outdoor)

All subjects in *type B* videos (truck simulator) were presented with the same track. Samples of **video #8** are shown in figure 3.15. Near-IR illumination was used and the *red-eye effect* is easily observable in the eyes of the subject. The driver wears glasses, and reflections appear in some frames (figure 3.15(c)). No illumination changes take place. There are some partial occlusions.

(a) (b) (c)

Figure 3.15: Samples of video #8

Sequence #	*Type*	Length (frames)	Gender	Glasses	Head turns	Motion blur	Gestures	Occlusions	Ill. changes	IR light
01	A	3895	Female	Yes	Yes	No	Yes	Yes	Slow	No
02	A	3748	Female	No	Yes	No	Yes	Yes	Slow	No
03	A	3500	Female	No	Yes	Yes	Yes	Yes	Fast	No
04	A	3300	Male	Yes	Yes	No	Yes	Yes	Fast	No
05	A	3887	Male	No	Yes	No	Yes	Yes	Fast	No
06	A	3400	Male	No	Yes	No	Yes	Yes	Fast	No
07	A	3417	Male	No	Yes	No	Yes	Yes	No	No
08	B	17100	Female	Yes	Yes	Yes	Yes	Yes	No	Yes
09	B	17376	Male	No	Yes	Yes	Yes	Yes	No	Yes
10	B	17413	Male	No	Yes	Yes	Yes	Yes	No	No
11	C	7261	Male	No	No	No	Yes	No	No	Yes
12	C	9079	Male	No	No	No	Yes	No	No	Yes
13	C	9091	Male	No	No	No	No	No	No	Yes
14	C	9091	Male	No	No	No	Yes	Yes	No	Yes

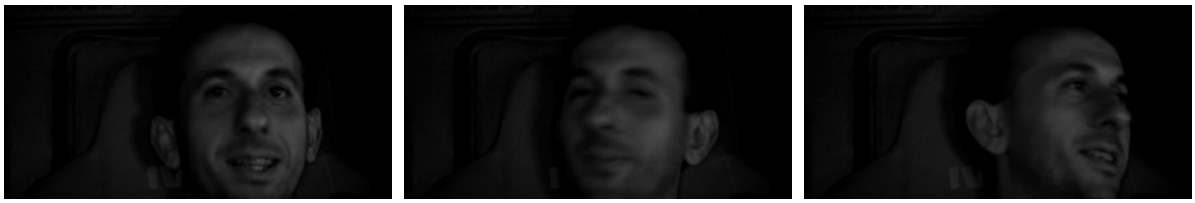
Table 3.1: Characteristics of the sequences in the RS-DMV dataset.

Figure 3.16 shows a few frames from **video #9**. Near-IR illumination was used, but no *red-eye effect* is observable, and the images are still very dark. The subject talks and moves his head frequently, around 30 times for the whole sequence. There are partial occlusions.

(a) (b) (c)

Figure 3.16: Samples of video #9

Figure 3.17 shows samples from **video #10**. Near-IR illumination was used, but no *red-eye effect* is observable. The subject talks and moves his head and eyes constantly and pronouncedly, more than 75 times for the whole sequence. There are partial occlusions during driving wheel turns, when the hands occlude part of the chin and mouth. Sample 3.17(b) is a case of motion blur.



(a) (b) (c)

Figure 3.17: Samples of video #10

Four *type C* videos contain subjects that exhibit signs of drowsiness, and at some point are close to fall asleep. All subjects were presented with the same track, although different from the track run at the truck simulator.

Images from *type C* videos will be shown pixelated in this work for privacy reasons. We are working towards obtaining the right of distribution for the sequences with these subjects.

A few shots from **video #11** and **video #12** are presented in figure 3.18. Even after pixelation, it is easy to notice that the subjects in the images are in fatigue, and about to fall asleep. Near-IR illumination was used in all four videos, although with low intensity.

3.3.3 Ground-truth data

Hand marking videos is a time consuming task, but it still is the best way of creating ground-truth data with minimal error. Videos in the RS-DMV database have been marked with 20 points, placed in specific positions of the face. The locations are the same as in the BioID set. Because of the length of the videos, not all frames have been marked. Points have been placed in the starting frames of all videos, and periodically after the initial seconds. The pattern is as follows.

- **Initial frames (0-10):** All frames from frame 0 to frame 10 have been marked.

(a) Sequence #11 (b) Sequence #11 (c) Sequence #11

(d) Sequence #12 (e) Sequence #12 (f) Sequence #12

Figure 3.18: Samples of videos #11 and #12

- **First seconds:** From frame 10 to frame 300 (second #10), 1 in 10 frames have been marked.
- **Rest of the video:** From second #10 until the end of the video, 1 in 30 frames (around 1 per second) has been marked.

The reasoning behind this pattern is to have a decent amount of samples for the initial seconds, so the performance of the algorithm can be roughly estimated without having to run it over the whole video sequence. This is helpful in cases where the algorithm has high processing times. Including a mark every 30 frames after the tenth second allows to monitor the performance, and detect if the algorithm fails to converge to the proper solution. Frames that have been marked are called *keyframes*.

A software tool, `Feature Point Marker` was developed to ease and speed up the marking of the images. This tool allows heavy reusing of marks placed in previous frames, and can handle occluded points. Please refer to the appendix A.1 for more details. The mean marking time per frame with 20 points is around 40 seconds, depending on the presence of occlusions and if a similar frame has been marked before.

3.4 Conclusions and contributions

In this chapter a short review of performance evaluation techniques has been presented. Also, some of the available image databases have been briefly discussed, their characteristics and interest for the work presented in this thesis. None of them have been found to represent the situations and actions that a system may find in real scenarios, and some of the problems that arise in those environments.

A new set of videos has been recorded to provide with enough data to test the algorithms in this thesis. They were recorded as part of various projects, with different objectives that complement each other. Videos recorded in a moving car outdoors show subjects in a real driving environment, constantly looking at rear-view mirrors and interacting with other passengers. Videos recorded in a truck simulator feature subjects that were presented with demanding driving environments. Finally, subjects in videos

recorded in a car simulator with a monotonous track to induce drowsiness show clear signs of fatigue and sleepiness.

While the amount of samples in this dataset is small compared with other databases (as the number of videos, not their frame count), it provides good examples of the actions and behaviors that appear in everyday driving. Selected frames in the videos in the database have been manually marked with 20 points, to be used as a reference for algorithm performance evaluation. A software tool was developed to ease the marking of the videos.

The main contribution of this chapter is the creation of a specific dataset for driving monitoring applications with one camera. The dataset can serve as the basis for assessing and comparing the performance of computer vision algorithms. To the best of our knowledge, it is the first video data set of this characteristics available in the state of the art. The RS-DMV dataset will be released to the research community in the near future.

Chapter 4

Active Models with *a priori* training

This chapter introduces active models, more precisely Active Shape Models (ASM) and Constrained Local Models (CLM) and evaluates their performance on the RS-DMV dataset.

Active Shape Models were introduced in [Cootes 95]. They are similar to the Active Contours (*snakes*) [Kass 88], but include additional constraints from a statistical shape model. Since their introduction, ASMs have been used widely in medical image processing, face expression recognition and many more applications. The shape is defined with a series of landmarks that fit locally to the best point in their surroundings, and then the constrains are applied: the shape model represents the subspace of all possible deformations the object can subject to, and the position of the landmarks is forced to belong in that subspace. This subspace of valid deformations is learned in advance from sample shapes of the object.

Researchers have extended the original ASM to include robust fitting [Rogers 02], boosted classifiers [Cristinacce 07] and many more. Other methods have been developed based on ASM, of which the Active Appearance Models (AAM) [Cootes 01a] are the best known. Instead of modeling the surrounding of the landmarks, the AAM models the appearance of the whole object. The points in the shape model (or Point Distribution Model) are meshed using a triangulation method (usually Delaunay's [Delaunay 34]) into a convex surface, and the appearance that falls in the triangles is characterized. This holistic approach to texture has problems when complex or changing illumination is present, as local changes in intensity are difficult to model and may lead to fitting errors.

Constrained Local Models (CLM) [Cristinacce 06] are closely related to AAM. Constrained Local Models try to increase the robustness to local illumination changes by working with small patches around the points in the shape. Even if the texture of all patches is modeled together, standard techniques like zero-mean normalized cross-correlation (ZNCC) can be used in pre-processing to remove illumination effects: if the patches are small enough, the changes will appear as an offset in all the pixels of the patch, and can be then removed easily.

Both Active Shape Model and Constrained Local Model have to be built in an offline training stage, from samples in a training set. Only variations included in the training set will be part of the model, and shapes and appearances left out of the model can not be recognized. Hundreds or thousand of images marked with landmarks are needed to learn a proper model. This task is usually done by a human operator, and can be very time consuming.

The performance of both ASM and CLM is tested with sequences from the RS-DMV dataset. The Stacked Trimmed ASM (STASM) [Milborrow 08] includes a number of improvements over the original ASM, and the authors released their source code under a

free license. This software, with a few modifications, has been used in the tests. We have written our own software implementing the CLM method based on the original paper, and use it to evaluate the performance of CLM.

The rest of this chapter is structured as follows. First the shape model used by ASM and CLM is formally presented. The Active Shape Model is introduced in 4.2. The texture modeling of CLM is discussed in 4.3. Section 4.4 presents the results obtained with ASM and CLM on the RS-DMV database. The chapter finishes with the conclusions in section 4.5.

4.1 Shape Model

In everyday language, *shape* is used to define the form or visual aspect of an object, more commonly referred to its outline. In a more mathematical form, shape is defined as [Dryden 98, Bookstein 91]:

Definition 4.1 ***Shape*** *is all the geometrical information that remains when location, scale and rotation effects are filtered out from an object.*

The usual way of describing a shape in the field of deformable models in an Euclidean space is by using *landmarks*, a finite number of points placed on the object. Landmarks are also called *key points*, *markers* or *fiducials* in the literature. Following [Dryden 98]:

Definition 4.2 ***Landmark*** *is a point of correspondence on each object that matches between and within populations.*

There are three types of landmarks, depending on the reason behind their placement.

Definition 4.3 ***Anatomical landmark*** *is a landmark that corresponds between organisms in a biologically meaningful way, such as the corner of a eye.*

Definition 4.4 ***Mathematical landmark*** *is a landmark located on an object according to some mathematical property, such as curvature or gradient.*

Definition 4.5 ***Pseudo-landmark*** *is a landmark placed between anatomical or mathematical landmarks, or over the outline of the object.*

The *configuration matrix* of the shape is the $k \times n$ matrix that contains the coordinates of the landmarks, where k is the number of landmarks and n is the dimensionality of the landmark space, usually 2 or 3. In this chapter, only bi-dimensional models are presented. Alternatively, a shape can be expressed as a $kn \times 1$ vector. For the 2D case,

$$\mathbf{s} = (x_1, y_1, x_2, y_2, \cdots, x_n, y_n)^t \tag{4.1}$$

In ASM and CLM, shapes are instantiated from a set of *shape vectors*, that without loss of generality are assumed to be orthonormal. With the *base shape*, these vectors control how the base shape deforms, and define the subspace of shapes that are considered valid. Let $\mathbf{p} = (p_1, \ldots, p_m)$ be the *shape parameters* for a particular shape $\mathbf{s}$, then

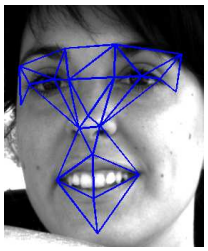
$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{m} p_i \cdot \mathbf{s}_i \tag{4.2}$$

where $\mathbf{s}_0$ is the base shape and $\mathbf{s}_i$ the shape vectors. Expressed as a matrix multiplication

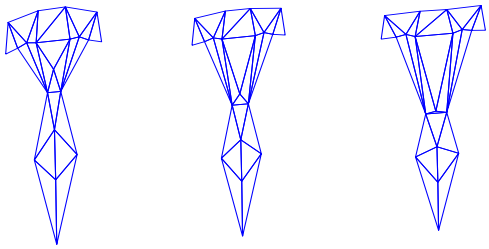
$$\mathbf{s} = \mathbf{s}_0 + \mathbf{S}\mathbf{p} \tag{4.3}$$

where $\mathbf{S} = (\mathbf{s}_1|\mathbf{s}_2|\cdots|\mathbf{s}_m)$.

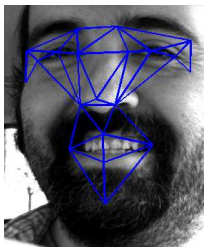
As the values of $\mathbf{p}$ change, the shape deforms. An example of a shape deformed along to different vectors is shown in figure 4.1. The central figures represent the base shape.



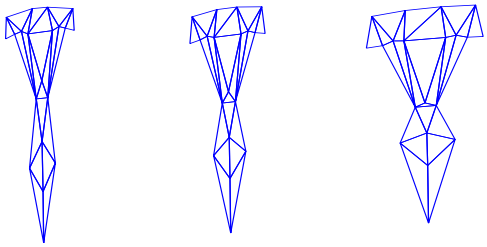
(a) Shape over a face



(b) Effect of the first vector: $p_1 = -k$, $p_1 = 0$ and $p_1 = +k$ respectively, $p_j = 0$, $\forall j \neq 1$



(c) Shape over a face



(d) Effect of the second vector: $p_2 = -k$, $p_2 = 0$ and $p_2 = +k$ respectively, $p_j = 0$, $\forall j \neq 2$

Figure 4.1: A shape deforming with two different vectors

The subspace of valid shapes is learned from a training set, that contains valid shapes, usually hand marked. Automatic model construction and landmark placement have received much attention from researchers [Cootes 05, Baker 04b], but they still do not obtain as good results as hand marked landmarks. In an Euclidean space, the Generalized Procrustes Analysis [Gower 75] is used to remove the scale, translation and rotation effects from the training set, resulting in a set of aligned shapes. The base shape $\mathbf{s}_0$ is set to be the mean of those shapes. No tangent space projection [Stegmann 02] is applied on the aligned shapes. These are well known methods, so they will not be presented here. Please refer to the literature for more details.

Principal Components Analysis (PCA) [Jolliffe 02] is performed on the aligned shapes, and the vectors that represent at least 95% of the accumulated deformation are chosen. Other dimensionality reduction techniques can be used instead of PCA, but it is still the most common method found in the literature.

This shape model is used by ASM, CLM and SMAT (see next chapter). The differences between these methods are in how they find the points in a new image that correspond to the landmarks.

4.2 Active Shape Model

In its original form, an Active Shape Model tries to locate the best position of a landmark using a normalized gradient vector over the perpendicular to the shape boundary at the current position of the landmark (known as *whisker*), as can be seen in figure 4.2. The gradient vector is called the *profile*. The mean profile vector $\mathbf{g}_0$ is obtained from the training set, as well as the covariance matrix $\mathbf{\Sigma_g}$.

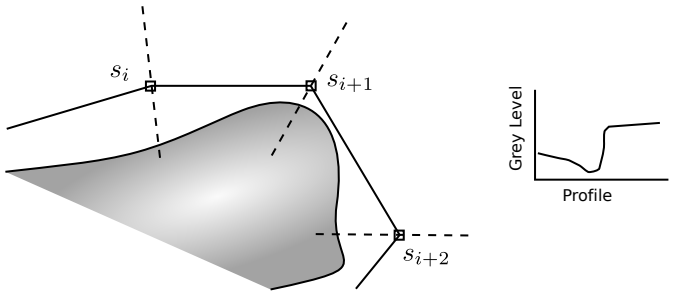


Figure 4.2: Whiskers of several shape landmarks

Searching is performed by moving the landmark over the *whisker*, and the position with the lowest Mahalanobis distance to the mean profile $\mathbf{g}_0$ is chosen as the new position.

$$\arg \min_{s_i' \in whisker} ((\mathbf{g}_i - \mathbf{g}_0)^T {\mathbf{\Sigma_g}}^{-1} (\mathbf{g}_i - \mathbf{g}_0)) \tag{4.4}$$

The shape model described above is used to constrain the positions of the landmarks, so they keep a meaningful form that do not degrades when one or more pixels fails in the search phase. Let $\mathbf{s}$ be the shape containing the position of the pixels after the search phase. If the shape vectors $\mathbf{s}_i$ were computed with PCA, they are orthonormal, and the parameters $\mathbf{p}$ can be obtained by projecting the shape $\mathbf{s}$ over each vector $\mathbf{s}_i$. This procedure gives the best estimate for the L_2 norm.

$$arg \min_{\mathbf{p}} ||\mathbf{s} - (\mathbf{s}_0 + \sum_{i=1}^{m} p_i \cdot \mathbf{s}_i)||^2 \tag{4.5}$$

$$\mathbf{p} = \mathbf{S}^T(\mathbf{s} - \mathbf{s}_0), \quad p_i = (\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{s}_i \tag{4.6}$$

The elements of $\mathbf{p}$ are usually thresholded to a maximum value. Too high values may distort the shape. The eigenvalues corresponding to the vectors represent the amount of variability in the training set over that vector. A scaled value of each eigenvector is used to limit the value elements in $\mathbf{p}$. This process of local fitting and imposition of shape constrains is repeated several times, until the shape ceases to move or other criteria is met.

The shape model in the ASM is independent from scale, rotation or translation, as they were removed in the training phase. In a 2D space, 4 extra parameters

$$\mathbf{t} = (x_t, y_t, \omega, k)^t \tag{4.7}$$

where ω is the angle and k is the scale, define a similarity transformation and are needed to represent any shape $\mathbf{s}$ that does not have a norm of value 1.

$$\mathbf{s} = T_{(x_t, y_t, \omega, k)}(\mathbf{s}_0 + \mathbf{Sp}) \tag{4.8}$$

For a single point $\mathbf{x}_j = (x_, y_j)^t$ of the shape $\mathbf{s}$, the transformation $T_{(x_t,y_t,\omega,k)}$ is applied as

$$T_{(x_t,y_t,\omega,k)} \begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} k\cos\omega & -k\sin\omega \\ k\sin\omega & k\cos\omega \end{pmatrix} \begin{pmatrix} x_j \\ y_j \end{pmatrix} + \begin{pmatrix} x_t \\ y_t \end{pmatrix} \tag{4.9}$$

The value of parameters $\mathbf{t} = (x_t, y_t, \omega, k)^t$ can be obtained using algorithm 4.1. The rotation and the scale are obtained by projecting the shape $\mathbf{s}$ over the mean shape $\mathbf{s}_0$ and $\mathbf{s}_0$ rotated $\pi/2$.

Algorithm 4.1 Estimate the parameters $\mathbf{t} = (x_t, y_t, \omega, k)^t$ from the current $\mathbf{s}$

```
1: procedure EstimateTransformation(s)
2:     (x_t, y_t)^t ← (1/n) Σ_{j=1}^{n} (s^j)        ▷ Mean of the points of s
3:     k cos ω ← s · s_0
4:     r_x ← −s_{0_y}                                 ▷ Rotate s_0
5:     r_y ← s_{0_x}
6:     k sin ω ← s · r
7: end procedure
```

The same level of independence from scale is desirable for the profiles, if they are to work successfully in a range of scales. Image pyramids are used on the training set images and profiles are learned for every level. The same procedure is done during fitting. Starting from the lowest level, the shape is fitted coarsely and subsequently refined every level until full image size. Algorithm 4.2 summarizes the steps of ASM fitting.

Algorithm 4.2 ASM fitting

```
1: procedure ASMFit
2:     L ← L_max                                      ▷ Start at the highest level of the pyramid
3:     while L ≥ 0 do
4:         repeat
5:             s ← T_{(x_t,y_t,ω,k)}(s_0 + Sp)
6:             for all points x_j = (x_j, y_j) do     ▷ Find best point in wishker
7:                 Find arg min_{x'_j ∈ whisker}((g_i − g_0)^T Σ_g^{-1} (g_i − g_0))
8:             end for
9:             Update p and t from the new s' = (x'_1, y'_1, ..., x'_n, y'_n)^t:
10:            (x_t, y_t, ω, k) ← EstimateTransformation(s')
11:            p ← S^T (T^{-1}_{(x_t,y_t,ω,k)}(s) − s_0)
12:        until convergence
13:        L ← L − 1
14:    end while
15: end procedure
```

4.2.1 Stacked Trimmed ASM

The stacked trimmed ASM (STASM) includes several extensions to the original ASM that have been developed since its introduction, and an evaluation of their effect on performance was presented in [Milborrow 08]. This extensions are:

- Increased number of landmarks. STASM tested the improvement in fitting error when the number of landmarks in a shape increased.

- Two dimensional profiles. Instead of a segment along the *whisker*, a square is placed around each landmark, increasing the search area. These are used on landmarks on the eyes and nose.

- Added noise in training. Gaussian noise was added to the shapes used in training. To simulate small head rotations, the left and right halves of the shapes were stretched alternatively.

- Increase the dimensionality of the shape model. The number of vectors $\mathbf{s}_i$ used in the search process is increased for the final iteration, when the fitting is expected to be more reliable. This extra vectors include additional modes of variation that may help improve the fitting.

- Trimming of profile covariance matrices. Removing non-significant values from the $\mathbf{\Sigma_g}$ matrices allows for faster computation of the Mahalanobis distance.

- Model stacking. Two models are used, the results for the first one are taken as the seed position for the second one. When the start shape is not well placed, using the second model improves the fitting.

STASM includes trained models that were built with samples from the XM2VTS [Messer 99] set that only include frontal faces. Because the shape model used in the classic ASM is linear, it has difficulties modeling non-linear effects such a face rotation around the vertical axis. That is, a rotation in 3D that appears as a non-linear deformation in 2D. The added noise and the stretching of the shapes in training allows the model to fit to slightly rotated faces, but it is not enough when the face turns noticeably, as is frequently the case in our test scenarios.

The STASM code is distributed under General Public License version 2 [FSF 91]. This code was modified to work on video sequences and produce results in the same form as the other pieces of software written for this thesis, i.e., movies and OpenCV XML files with the numerical results of the model fitting.

4.3 Model fitting in CLM

Constrained Local Model takes a different approach to search for the best position of the shape's landmarks. Instead of using the mean profile along the *whisker* and the covariance matrix, the texture of a patch around each landmark is modeled with another vector base, obtained using PCA.

Each patch E_i is extracted from the images in the training set and normalized to zero mean and unit variance. All normalized patches from an image are concatenated as $\mathcal{E} = (E_1|\dots|E_n)$ and then reshaped into a 1D vector. A matrix is formed with a vector from each sample as a row. PCA is applied on the matrix, from which the mean texture $\mathbf{A}_0$ and vectors $\mathbf{A}_j$ are obtained. A new set of (concatenated) templates can be instantiated using this vector base and a set of coefficients $\mathbf{w}$. The range of values that $\mathbf{w}$ can take are restricted, as was the case with the shape parameters $\mathbf{p}$ above.

$$\mathbf{A} = \mathbf{A}_0 + \sum_{j=1}^{s} w_j \cdot \mathbf{A}_j \tag{4.10}$$

In the original proposal [Cristinacce 06], shape and texture parameters are coupled in another set, on which PCA is applied again to reduce its dimensionality. Our implementation, however, keeps the two parameter sets independent.

A block diagram of the CLM fitting process is presented in figure 4.3. The process is described in algorithm 4.3.

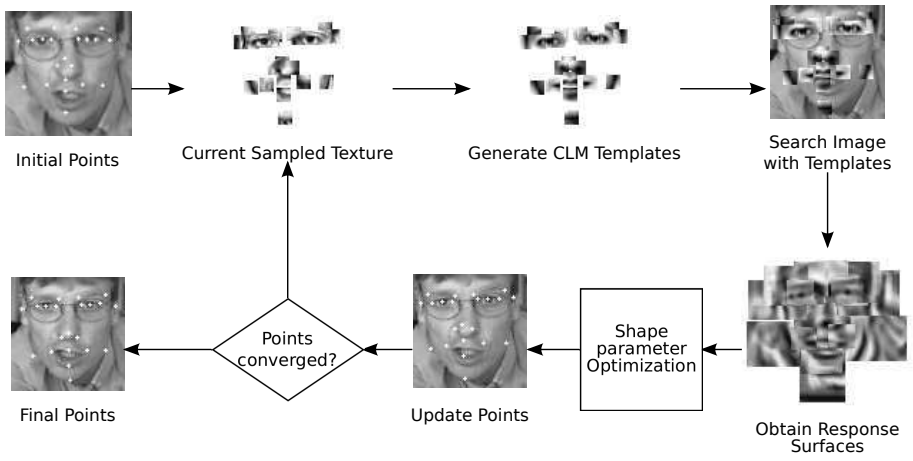


Figure 4.3: CLM fitting algorithm. Image from [Cristinacce 06]

Algorithm 4.3 CLM fitting

```
1: procedure CLMFit
2:     repeat
3:         s ← T_(x_t,y_t,ω,k)(s_0 + Sp)
4:         A ← A_0 + Σ_{j=1}^{s} w_j · A_j
5:         Reshape A into E = (E_1|...|E_n)                ▷ A is 1 × N
6:         for all points x_i do
7:             Compute the response R_i(x_i, y_i), correlating E_i around the position s^j
8:         end for
9:         Minimize f(q) = Σ_{i=1}^{n} R_i(x_i, y_i) + K Σ_{j=1}^{m} −p_j^2 / λ_j
10:        Obtain the new values of p, q and w.
11:    until convergence
12: end procedure
```

From the current values of $\mathbf{w}$ and $\mathbf{p}$, the location of the landmarks $\mathbf{x}_i$ of the shape $\mathbf{s}$ and texture $\mathbf{A}$ are obtained. Each patch is used to compute a response surface $\mathbf{R}_i(x_i, y_i)$ over the search image. The surface is obtained by correlating the model templates in an area around the corresponding landmark $\mathbf{x}_i$. The function to optimize is $f(\mathbf{q})$, with $\mathbf{q} = (\mathbf{p}^t, \mathbf{t}^t)^t$, which balances the best response surfaces with the values of the shape parameters:

$$f(\mathbf{q}) = \sum_{i=1}^{n} \mathbf{R}_i(x_i, y_i) + K \sum_{j=1}^{m} \frac{-p_j^2}{\lambda_j} \tag{4.11}$$

where λ_j are the eigenvalues and K is a constant. The second term is an estimation of the log-likelihood of the shape given its parameters and eigenvalues (see [Cristinacce 08] for more details). The optimization is done with the Nelder-Mead simplex algorithm [Nelder 65].

4.3.1 Initialization and tracking losses

In Cristinacce's work, the model position is initialized using first Viola & Jones face detector, and then smaller detectors using the same algorithm, constrained with the Pictorial Structure Matching of [Felzenszwalb 05]. We have found that the STASM implementation provides accurate localization when the face is frontal (see the results below, and in the original paper), and then use it to initialize the CLM position. Although STASM does not run in real-time, a one-time delay is acceptable for our intended application.

Automatic tracking loss detection during execution is of great importance, as it provides an extra assurance that the model is at least tracking the face, regardless of the magnitude of the error, and has not diverged. Two simple checks were employed:

1. An scale and orientation log is maintained. If the shape rotates around itself or shrinks, the previous check will not detect it. The values of the rotation and scale parameters (ω, k) of the shape are recorded. The allowed rotation range is ± 50 degrees. The scale is modeled as a Gaussian. After an initialization period, if the scale k at any moment is more than 2 standard deviations away from the mean, the model is re-initialized. This check is valid because a driver stays at an approximately constant distance from the driving wheel.

2. Viola & Jones face detector is executed periodically, and the number of landmarks that fall inside the bounding box counted. If they are below a threshold, the model is re-initialized.

The block diagram in figure 4.4 represents this process.

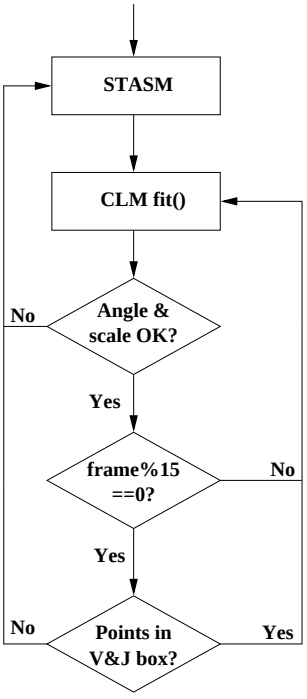


Figure 4.4: CLM execution graph, with tracking loss detection

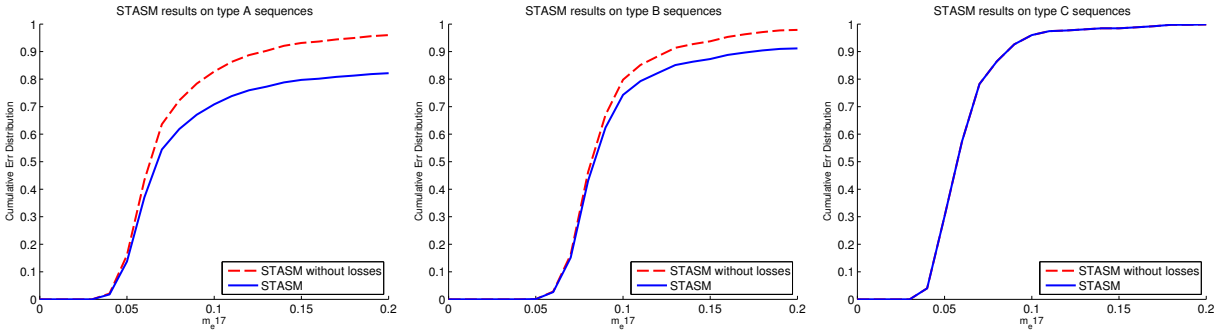
4.4 Results

This section presents the results of fitting CLM and STASM to the sequences in the RS-DMV. Both algorithms were run over the sequences in the database. STASM code was written to work on single images, and an interface to the code was written to work on sequences of images. For each frame, a face detector (we selected Viola & Jones) is used to locate the face, and from there the model is fitted. Viola & Jones does not always succeeded, in that case the position of the model in the previous frame was used to initialize the bounding box in the current frame. This proved to be a simple solution that worked in most failures of the face detector.

The results presented below represent only the error values obtained when the ground-truth position of the points was available, 1 in every 30 frames (approximately 1 per second). As such, events can take place in less than a second may not reflect on the results: quick hand occlusions, fast head turns or illumination changes. In some cases, these events lead to a tracking loss, that may not be detected or recovered when the next *keyframe* is processed. For the CLM, when a tracking loss is detected in a *keyframe* or the previous one, the model is reinitialized, but the frame is marked as a loss in any case. This avoids taking the fitting results of STASM as CLM's. Frames with tracking losses are assigned a fitting error value of $m_e17 = 10$.

Full resolution images have not been used. A size of 320×240 pixels was found to be enough for the methods, and all frames were subsequently resized. Aspect ratio was kept, so depending on the original size, the dimension of the images were a bit different.

STASM performed with low error when the face is frontal, but can not fit to the face during pronounced head turns. Figure 4.5 shows the m_e17 of STASM for the 3 different types of sequences. Overall results are drawn in solid line, while results without measurements corresponding to tracking losses are drawn with a dotted line.

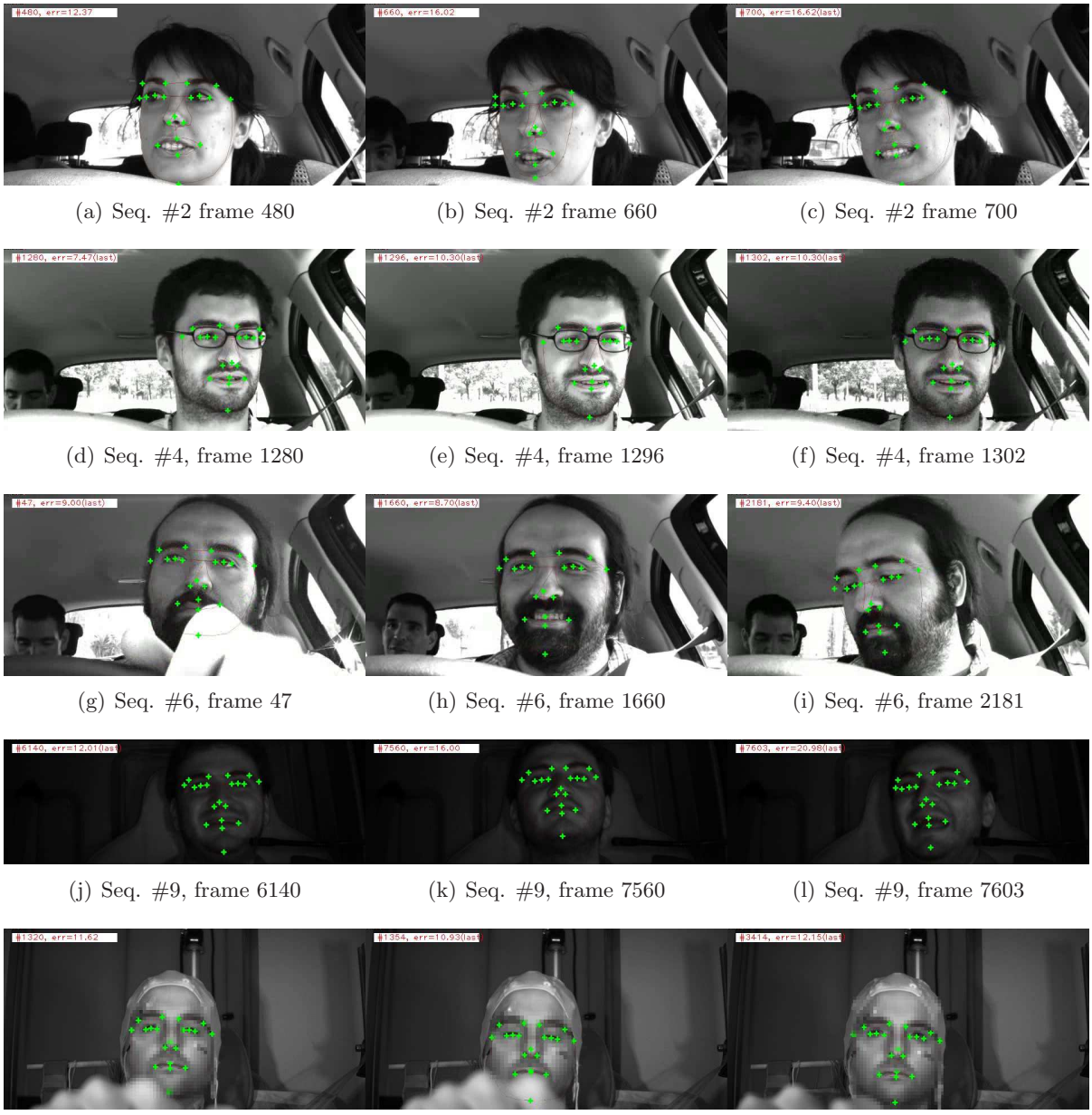


(a) *Type A* (b) *Type B* (c) *Type C*

Figure 4.5: STASM cumulative distribution of the m_e17 measure for the 3 types of sequences

Figure 4.5(a) shows that *Type A* videos pose a greater challenge to STASM, because they contain frequent movements, continuous talking and occlusions. In *Type C* videos the person moves little and there are few or no head turns. STASM obtains very good results in this case. Results for *Type B* videos stay in a middle ground. Figure 4.6 shows some captures of several videos with the model fit.

The Constrained Local Model was tested on the same sequences. Several model parameters were used. CLM, as introduced in [Cristinacce 06] does not implement a multi-scale approach. When the size of the faces in the training and testing set are roughly the same and the initialization is accurate in scale, a one-scale approach is enough. However, when



(a) Seq. #2 frame 480 (b) Seq. #2 frame 660 (c) Seq. #2 frame 700

(d) Seq. #4, frame 1280 (e) Seq. #4, frame 1296 (f) Seq. #4, frame 1302

(g) Seq. #6, frame 47 (h) Seq. #6, frame 1660 (i) Seq. #6, frame 2181

(j) Seq. #9, frame 6140 (k) Seq. #9, frame 7560 (l) Seq. #9, frame 7603

(m) Seq. #11, frame 1320 (n) Seq. #11, frame 1354 (o) Seq. #11, frame 3414

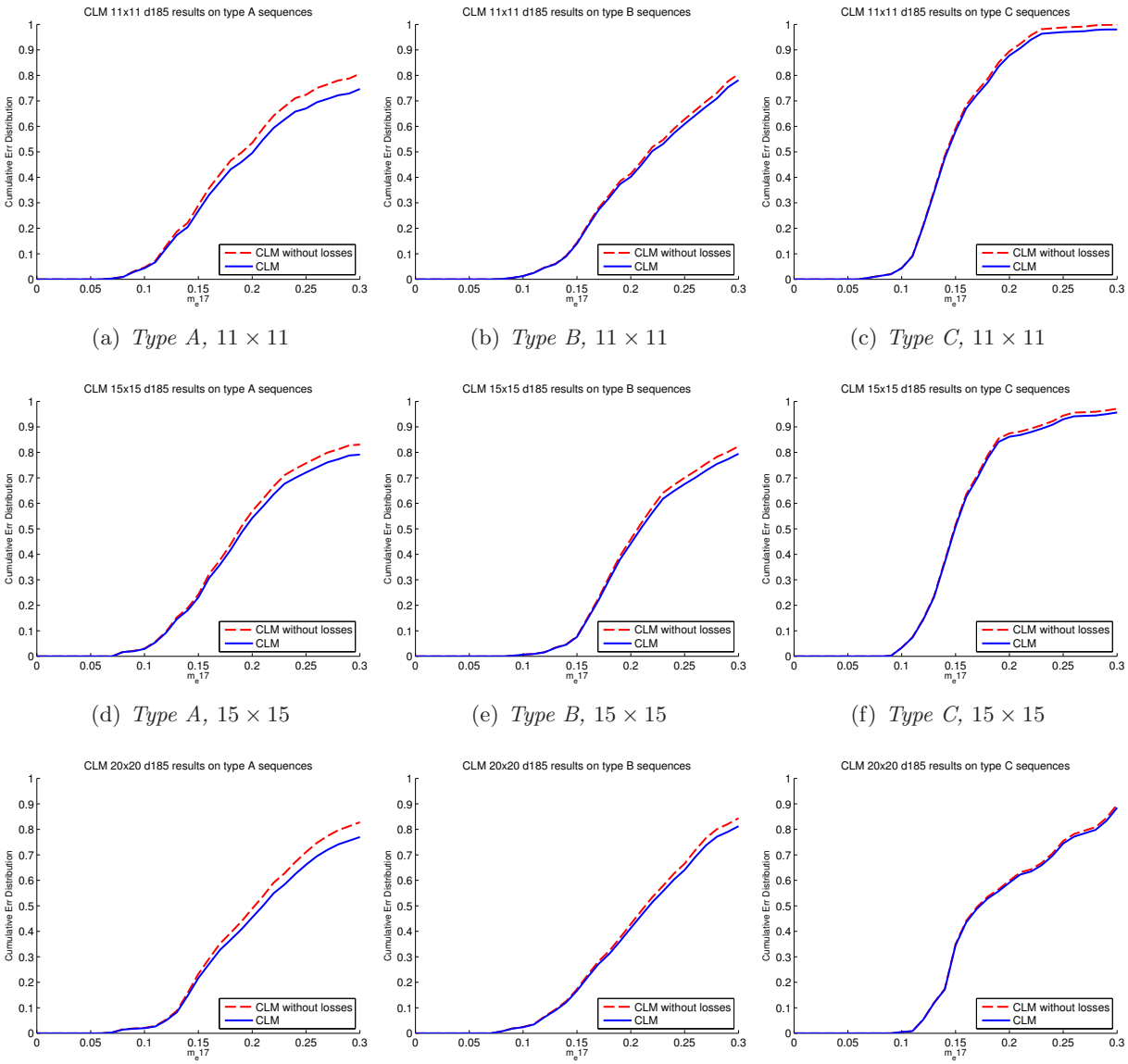
Figure 4.6: Frames of several videos with STASM fit

the size of the faces in the images in the training and test sets differs ostensibly, a scale correction may be needed so the facial features and the texture correspond correctly. The distance between the eyes in the images in the training set (the BioID database) was used as the reference. The mean value of the reference for the images in that database is around 185 pixels, and thus this value was chosen to minimize the number of images that need to be scaled. To provide a comparison, another model was trained without performing any escalation on the images. The first model is noted as *d185*, and the latter *d0*.

Different patch sizes have been tested: 11×11, 15×15 and 20×20 pixels. 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. Bigger patches are easier to

locate, but are less specific and *non-feature* elements may appear in them: with sufficient size, a patch centered around the eye may include part of the eyebrow, which may not be desirable. All patches in the model are square and of the same size.

Thus, 6 different configurations were tested for CLM. Figure 4.7 shows the cumulative distribution of m_e17 of CLM using scale correction *d185*. Results obtained without scale correction (*d0*) are shown in figure 4.8



(a) *Type A*, 11×11 (b) *Type B*, 11×11 (c) *Type C*, 11×11

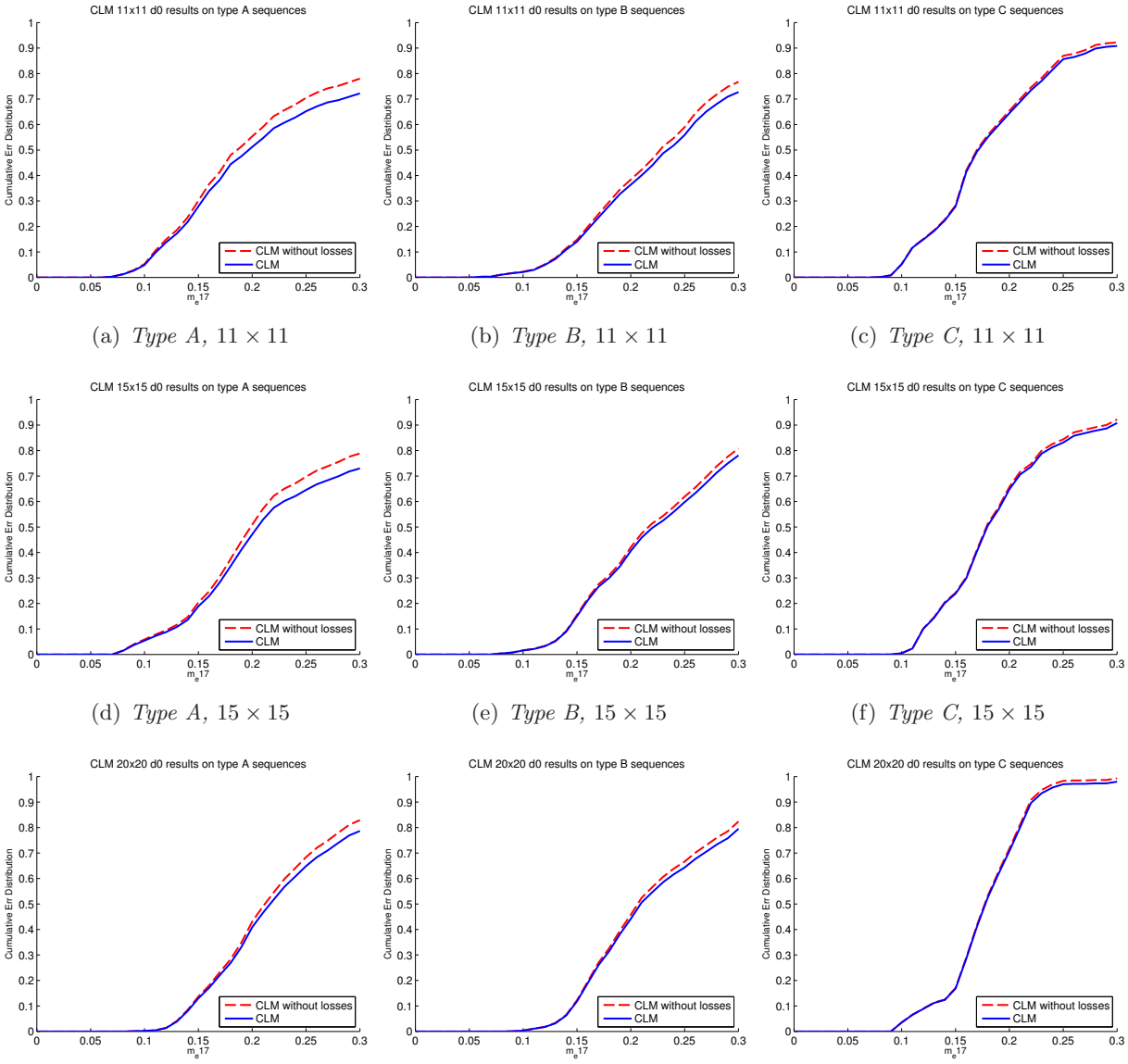
(d) *Type A*, 15×15 (e) *Type B*, 15×15 (f) *Type C*, 15×15

(g) *Type A*, 20×20 (h) *Type B*, 20×20 (i) *Type C*, 20×20

Figure 4.7: CLM cumulative distribution of the m_e17 measurement with scale correction and different patch sizes

Figures show that results for all CLM configurations are poor, with results from *type C* sequences being the best. There are several reasons that might explain this. Because in *type C* sequences the person does not move much, the fitting error may come from differences between the appearance of the patches in training set and on the test set. Contrary to STASM, results for *Type B* videos do not see an improvement over *Type A*'s.

CLM was trained with samples from the BioID database, whose images only contain



(a) *Type A*, 11×11 (b) *Type B*, 11×11 (c) *Type C*, 11×11

(d) *Type A*, 15×15 (e) *Type B*, 15×15 (f) *Type C*, 15×15

(g) *Type A*, 20×20 (h) *Type B*, 20×20 (i) *Type C*, 20×20

Figure 4.8: CLM cumulative distribution of the m_e17 measurement without scale correction for different patch sizes

faces in upright position. In some sequences of the RS-DMV dataset the head of the driver is at moments leaning to the left or right, and laying on the head rest. A small rotation reduces the value of the correlation quickly, making an accurate fitting more difficult. The slightly off-center position of the cameras adds to this problem. Another reason behind the error values is that the images are dark, and normalization may not be enough to bring pixel levels to correspond with those the model can synthesize. Also, with many pixels being of similar levels, the standard deviation may be very close to zero, amplifying noise to dominant values when the standard deviation is forced to be the unity. For *type A* sequences, movement and face pose are the dominant source of errors. In other situations where there are no occlusions or movements, unsuccessful fitting in few points in CLM lead to high error values and drifting.

There are no significant improvements between the different configurations. For models

with scale correction, the model with patches of 15×15 pixels obtains slightly better results than the rest. The differences between results using scale correction or not are not remarkable, but models with scale compensation work a little better, specially the ones with 11×11 and 15×15 patch sizes. The model with 20×20 patches obtains better results without escalation, although the differences are small.

The results obtained for CLM are much poorer than those reflected by Cristinacce in [Cristinacce 08] for a sequence of a driver. Because both the implementation and the test data are different, and the implementation and sequence of Cristinacce are not available, it is not possible to draw a conclusion.

From now on, the results shown for CLM refer to the configuration of 15×15 pixels with scale correction. A few samples of videos with this configuration of CLM fit are shown below in figure 4.9. The sequences and frames are the same as in figure 4.6.

Table 4.1 shows the rate of tracking losses of STASM and CLM for *Type A*, *B* and *C* sequences, the maximum rate of losses in a sequence and the minimum. These numbers only refer to the percentage of *keyframes*. With only a exception, STASM shows a lower number of tracking losses that CLM, which indicates a more robust algorithm.

		Mean	Maximum	Minimum
STASM	*Type A*	5.43%	15% (seq. #5)	0%(seq. #4)
	Type B	2.54%	3.27%(seq. #10)	1.8%(seq. #9)
	Type C	0%	0%	0%
CLM	*Type A*	7.64%	13.75%(seq. #1)	2.08%(seq. #6)
	Type B	4.26%	6.54%(seq. #10)	0.64%(seq. #9)
	Type C	1.5%	2.4%(seq. #12)	0.37%(seq. #11)

Table 4.1: STASM and CLM track losses for different types of sequences

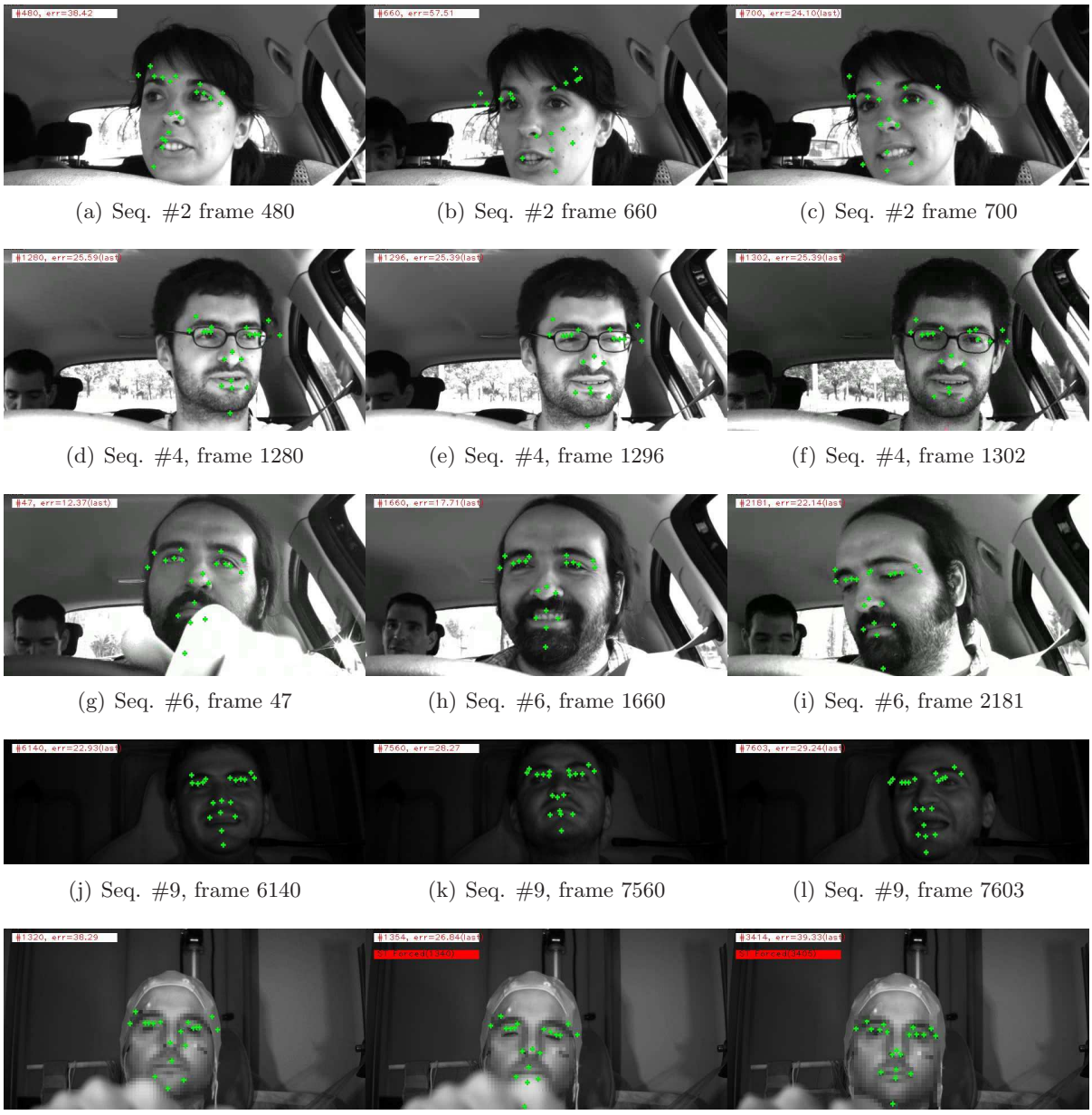
4.4.1 Processing times

Processing times for STASM and CLM were measured. Table 4.2 shows the average frames per second that the algorithms can process, the standard deviation and the worst case. With an average 2.2 fps, STASM processing time is far away from the 30fps required for the application of this thesis. This figure does not include the time employed by the Viola & Jones face detector that STASM uses for initialization. Milborrow and Nicolls report lower search times, around 5 fps (0.2 seconds per frame), including face detection, on a 3GHz Pentium. Our tests were run on a Intel Xeon 2.2 GHz (without multi-threading in any case) running GNU/Linux with GCC 4.2 as compiler, and optimizations disabled (`-O0`). Different compilers and compiler options may also explain the different results, which are in any case far from real-time execution.

All 6 configuration of CLM run with similar computing times, at more than 30 frames per second. Processing times are more stable in STASM than CLM.

Configuration	Mean (fps)	Sdv (fps)	Worst frame (fps)
STASM	2.17	0.13	1.96
CLM, 15×15	33.07	5.36	23.14

Table 4.2: Execution time for SMAT



(a) Seq. #2 frame 480 (b) Seq. #2 frame 660 (c) Seq. #2 frame 700

(d) Seq. #4, frame 1280 (e) Seq. #4, frame 1296 (f) Seq. #4, frame 1302

(g) Seq. #6, frame 47 (h) Seq. #6, frame 1660 (i) Seq. #6, frame 2181

(j) Seq. #9, frame 6140 (k) Seq. #9, frame 7560 (l) Seq. #9, frame 7603

(m) Seq. #11, frame 1320 (n) Seq. #11, frame 1354 (o) Seq. #11, frame 3414

Figure 4.9: Frames of several videos with CLM fit

4.5 Conclusions

This chapter has presented the ASM and CLM techniques, and has tested them using the video sequences in the dataset. The STASM implementation of ASM was used to test its performance. An implementation of CLM was written and tested, using various configurations of parameters. STASM demonstrates good performance when the face is frontal to the image, and is more robust in low-light situations. Although robust estimators are not used, STASM uses 68 points to fit the model, which improves robustness and reduces the fitting error. It also uses a multi-scale approach. Mean processing time for STASM is close to 0.4 seconds per frame.

Several configurations of CLM were tested with different patch sizes and rescalation of

the images so the faces in them had a similar size. In general, CLM's results are poorer than expected, and it requires frequent reinitialization (which is done with STASM in our implementation). Best results are obtained with patches of 15×15 pixels, but the improvement over other sizes is not remarkable. Low light situations seem to have a negative effect on CLM, leading to frequent errors in tracking. Unlike STASM, CLM does not have any feature to enhance its robustness, and erroneous fitting of a few patches may lead to an improper fit of the model. On a positive side, CLM is able to run in real time.

The common part between STASM and CLM is the shape model. Because STASM has good accuracy, the shape model is able to synthesize shapes with landmarks coincident or close to the real position of the landmarks in the ground-truth. STASM fitting is slow, and the appearance model of CLM has problems tracking patches of rotated or leaning faces, so it is reasonably to look for solutions or alternatives to the appearance modeling of CLM. One possibility is to modify the model to one that instead of normalized gray levels uses characteristics that are less sensitive or invariant to rotations. Another is to follow an solution similar to Simultaneous Modeling and Tracking (SMAT), building a model incrementally as the face moves and rotates, and incorporating this variations to the model to improve fitting next time they take place. The possibilities of this approach are explored in the next chapter.

Chapter 5

Simultaneous Modeling and Tracking

This chapter introduces the Simultaneous Modeling and Tracking (SMAT) method, and presents its most important characteristics. SMAT was introduced by Dowson *et al.* in [Dowson 05]. It is closely related to Constrained Local Models, but its main feature is that it does not require any previous training. Instead of fitting a pre-learned model to an object in a sequence of images, SMAT builds the model as the object changes in the images, while tracking it. This method is not able to work on static images, but this is not required by the application.

The reasoning behind SMAT is that obtaining *a priori* models 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. On the other hand, unless all possible variations in the object class are known and the training set covers all of them, the models will be incomplete. While most of the objects in the class will be correctly recognized and tracked in most situations, least common objects will be not.

Several methods that work without *a priori* models have been presented in the literature. Most of them focus on patch tracking on a video sequence. The classic approach is to use the image patch extracted on the first frame of the sequence to search for similar patches on the following frames. Lukas-Kanade method [Lucas 81] was one of the first proposed solutions and it is still frequently used. This algorithm uses Newton optimization method to find the best matching patch.

More recent approaches include works that relay in more complex modeling of the patch to increase the robustness and precision of the tracking. Jepson *et al.* [Jepson 03] presented a system with appearance model based on three components: a stable component that is learned over a long period based on wavelets, a 2-frame tracker and an outlier rejection process. An EM-algorithm is used to adapt the model parameters. 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. Yin and Collins [Yin 07] build an adaptive view-dependent appearance model on-line. The model is made of patches selected around Harris corners. Model and target patches are matched using correlation, and the change in position, rotation and scale is obtained with the Procrustes algorithm.

Another successful line of work in object tracking without *a priori* training is based on classification instead of modeling. Collins and Liu [Collins 05] presented a system based on background/foreground discrimination. This system runs in real time, but it is only able to track objects bounded by a rectangle. Avidan [Avidan 07] presents one of the many

systems that use machine learning to classify patches [Grabner 06, Pham 07b]. Avidan uses weak classifiers trained every frame and AdaBoost to combine them. A confidence map is generated from the classified pixels and mean-shift is used to localize the object on the map. Pilet *et al.* [Pilet 05] train keypoint classifiers using Random Trees that are able to recognize hundreds of keypoints in real-time.

SMAT is in line with the former approach, building a model and relaying on matching to track patches. Methods as Lukas-Kanade work reliably while the appearance of the patch in the current frame is similar to the template extracted at the beginning of the sequence. However, if its appearance changes, the distance between patch and template will be big enough to cause an incorrect localization, and the tracking will fail. Updating the template using the previously tracked patch is a simple alternative, but it accumulates matching errors over time and drifts from the intended object over time. Matthews *et al.* [Matthews 04a] proposed an *strategic update* of the template, that keeps the template from the first frame to correct errors that appear in the localization. When the error is too high, the update is blocked. This approach obtains good results, but fails if the appearance changes enough to repeatedly ban the update. A typical source of permanent changes is relative movement. In [Segvic 06], a solution is proposed where the template is not updated, but the window around the selected features is adaptively detected and selected.

These methods have a strong dependence on the quality of the template extracted in the first frame, a bad choice will make them fail very quickly. SMAT was developed as an extension to them, and tries to solve the problems of representation and drifting using a more complex model. SMAT has its own weaknesses for the intended application of this thesis, and we propose a more robust algorithm, called Robust SMAT (R-SMAT).

The rest of the chapter is structured as follows. The original proposal by Dowson *et al.* is presented in section 5.1. Section 5.2 discusses the algorithm. Sections 5.3 and 5.4 present some modifications to address several of its weaknesses, and introduces Robust SMAT. Tests and results of the performance of the approaches are presented in section 5.5. Conclusions and contributions close this chapter.

5.1 Simultaneous Modeling and Tracking

Simultaneous Modeling and Tracking (SMAT) was first proposed by Dowson *et al* in [Dowson 05, Dowson 06]. As mentioned above, the main purpose of SMAT is to track an object in an image sequence without a pre-learned model. Tracking an object can be approached by tracking the object as a whole or just some of its features. When tracking several features, relations between their positions can be learned and enforced as part of the process, as was done in Constrained Local Models. SMAT tries to model both the appearance of the features and how their positions are related. Feature position is known as *structure*, and a *structure model* is used. For consistency with previous chapters, the nomenclature *point distribution* or *shape* will be used instead.

Both the appearance and point distribution models are independent. In a first stage, object features are tracked with their respective appearance models. Their final positions are then processed using the point distribution model. If the positions are considered reliable and not caused by tracking errors, the appearance model is updated, otherwise it is left unchanged. Figure 5.1 shows a flow chart of the algorithm.

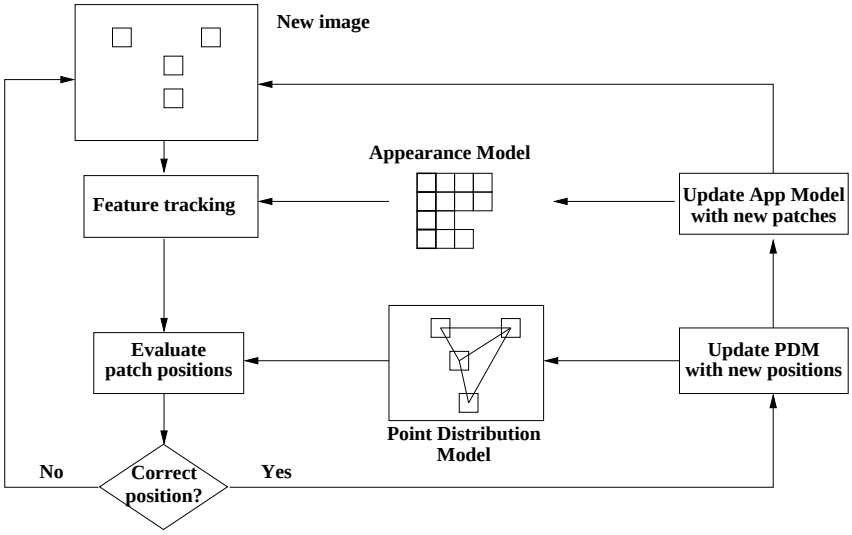


Figure 5.1: SMAT block diagram

5.1.1 Appearance Modeling

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

SMAT builds a library of exemplars obtained from previous frames, image patches in this case. Dowson *et al.* assumed that the points in each cluster in feature space follow a Gaussian distribution, and clusters can be defined by their median and variance. Using the median can result in a cluster with a greater variance that than obtained using the mean. However, the mean results in a smoothed patch that may not correspond to any possible appearance. More importantly, the median is more robust to outliers that the mean.

The clusters are updated incrementally as new images become available. The membership $m_k(x)$ of a new patch to a cluster k will depend on the relative distance to the median (or *representative*), and a threshold relative to the variance of the cluster. While this would result in fuzzy clustering, a hard threshold is used:

$$m_k(x) = \begin{cases} 1 & \mathrm{d(x}, \mu_\mathrm{k}) < \tau(\sigma_\mathrm{k}) \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where μ_k and σ_k are the median and the standard deviation of cluster k. In a later work, Dowson *et al.* [Dowson 06], introduced a different condition for membership, that compares the probability of the exemplar belonging to foreground (a cluster) or to the background

$$\frac{p(fg\,|d(x,\mu_n),\sigma_{fg_n})}{p(bg\,|d(x,\mu_n),\sigma_{bg_n})} \tag{5.2}$$

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

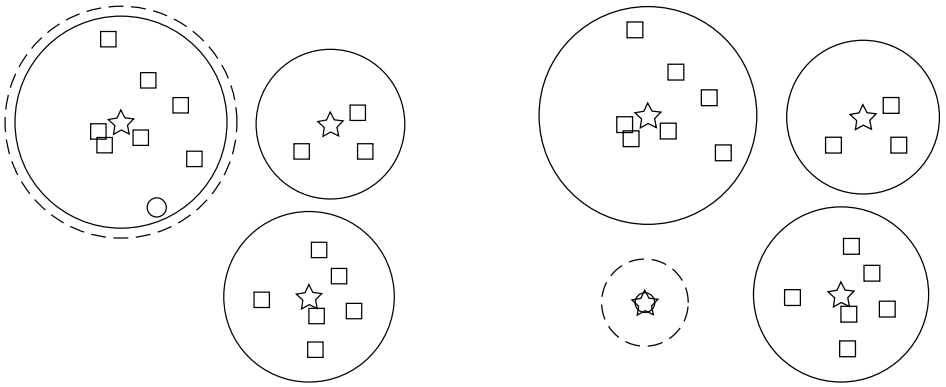
Under the condition in equation 5.1 a patch could belong to two or more clusters. To avoid this, clusters are given a weight that depends on the frequency of patches being included in the cluster. If a new patch belongs to two or more clusters, it is added to the one with the biggest weight. For each new frame, the weight is updated as

$$w_k^{(t+1)} = \begin{cases} (w_k^{(t)} + \alpha)\frac{1}{1+\alpha} & if \quad k = k_u \\ w_k^{(t)}\frac{1}{1+\alpha} & \text{otherwise} \end{cases} \tag{5.3}$$

where $\alpha \in [0, 1)$ is the learning rate, and k_u is the index of the updated cluster. If the new patch is not within the threshold of any cluster, a new cluster is created, with the new patch as its only member. The weight of this new cluster is initially set to zero

$$w_k^{(t+1)} = 0 \qquad if \quad \text{cluster } k \text{ is new} \tag{5.4}$$

A 2D example can be seen in figure 5.2. In 5.2(a), the exemplar is included in one of the clusters. The variance σ_{fg} of this cluster increases slightly, and so does the inclusion threshold. In the other case, 5.2(b), a new cluster is created and the patch become the representative of that cluster. Note that the membership threshold is the same for all dimensions.



(a) Inclusion of a new exemplar in a cluster

(b) Creation of a new cluster

Figure 5.2: Incremental clustering of exemplars. Pre-inclusion threshold is shown as solid line, post-inclusion threshold is shown as a dotted line. The stars indicate the representative of the clusters, and the circle the new exemplar.

The median of each cluster is obtained by constructing a matrix of distances between the elements (patches) $E_1, \ldots, E_N$ in the cluster, as in equation 5.5.

$$\begin{pmatrix} d(E_1, E_1) & d(E_1, E_2) & \ldots & d(E_1, E_N) \\ \vdots & \ddots & & \vdots \\ d(E_N, E_1) & d(E_N, E_2) & \cdots & d(E_N, E_N) \end{pmatrix} \tag{5.5}$$

Matrix columns are summed, and the exemplar corresponding to the column with the lowest sum is chosen as the median. This matrix can be constructed incrementally as patches are added to the cluster.

To limit memory requirements, the number of patches in a cluster is limited to a maximum of M, as is the number of clusters, K. When M is reached for a cluster, the most distant patch from the median is removed. This patch has the highest probability

of being an outlier to the cluster, not being a variation of the same appearance as the representative. Likewise, when the number of clusters reaches K, the cluster with the lowest weight w_k is discarded. This way, the model removes clusters that have seen less updates lately.

The original proposal of SMAT used Mutual Information (MI) as a distance measure to compare the image patches, and found it to perform better that Sum of Squared Differences (SSD), and slightly better than correlation in some tests. Any definition of distance can be used. We have also tested Zero-mean Normalized Cross-Correlation (ZNCC). Several types of warping were tested in [Dowson 06]: translation, euclidean, similarity and affine. The results showed an increasing failure rate as the degrees of freedom of the warps increased. Based on this, we have chosen to use the simplest, and the patches are only translated depending on the point distribution model.

5.1.2 Point Distribution Model

SMAT point distribution model is built in a similar fashion to that of the appearance. Clusters are formed with shapes that are close in the point distribution space. The mean of the shapes in each cluster is used as representative. Mahalanobis distance is used to compare the shape resulting from the independent feature tracking with each representative:

$$d_k(\mathbf{s}) = [(\mathbf{s}' - \bar{\mathbf{s}}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{s}' - \bar{\mathbf{s}}_k)]^{\frac{1}{2}} \tag{5.6}$$

where $\mathbf{s}'$ is the shape $\mathbf{s}$ translated to the origin, $\bar{\mathbf{s}}_k$ is the representative of cluster k also translated to the origin, and $\mathbf{\Sigma}_k$ is the covariance matrix of the data in the cluster. As $\mathbf{\Sigma}_k$ will often be non-invertible, singular value decomposition (SVD) is used to obtain its eigenvalues $\mathbf{D}$ and eigenvectors $\mathbf{A}$, and the distance is computed as

$$d_k(\mathbf{s}) = [\mathbf{A}(\mathbf{s}' - \bar{\mathbf{s}}_k)]\mathbf{D'}^{-1} \tag{5.7}$$

where $\mathbf{D'}^{-1}$ is the pseudo-inverse of $\mathbf{D}$.

The Mahalanobis distance is normalized to the standard deviation of the data, and thus the membership threshold can be set to any number τ_k, that will correspond to the number of standard deviations. Following [Dowson 05], we have chosen

$$\tau_k = 1 \tag{5.8}$$

5.1.3 Enforcing point distribution model constraints

Once the shape $\mathbf{s}$ has been found to belong to a cluster k, the positions are constrained by projecting the shape over the eigenvectors of the cluster and limiting the coefficient values to the eigenvalues in $\mathbf{D}$. The constrained shape $\mathbf{s}'_c$ is computed as

$$\mathbf{s}'_c = \mathbf{A} min(\mathbf{A}^T(\mathbf{s}' - \bar{\mathbf{s}}_k), \mathbf{D}^{-\frac{1}{2}}) + \bar{\mathbf{s}}_k \tag{5.9}$$

If the distance between the position of a feature in $\mathbf{s}'$ and $\mathbf{s}'_c$ is greater than δ, the appearance model of this feature is not updated to prevent introducing outliers. Also, in the next frame the feature position is reset to its value in $\mathbf{s}'_c$.

In the case that the shape S does not belong to any existing cluster, a new one is created in the point distribution model only if less than 25% of the feature trackers have created new clusters in their models [Dowson 05]. Introduction of novel data in both

types of models at the same time indicates a possible tracking loss, and it is blocked for increased robustness.

The fitting process of SMAT uses these models of texture and shape, and is summarized in algorithms 5.1 and 5.2.

Algorithm 5.1 SMAT fitting

Let $C_j = \{\mathcal{C}_{j_1}, \ldots, \mathcal{C}_{j_n}\}$ be the set of n clusters that model the texture around $\mathbf{x}_j$, with weights $\{w_{j_1}, \ldots, w_{j_n}\}$ and membership thresholds $\{T_{j_1}, \ldots, T_{j_n}\}$

procedure SMATFIT
  for all points $\mathbf{x}_j$ in $\mathbf{s}$ **do**
    for all cluster $\mathcal{C}_{j_k}$ in C_j **do**
      Find region P around $\mathbf{x}_j$ that minimizes $d(\mathcal{C}_{j_k}, P)$
      if $d(\mathcal{C}_{j_k}, P) < T_{j_k}$ **then**
        $P_j \leftarrow P$ ▷ Save P
        $\mathbf{x}'_j \leftarrow centreOf(P)$ ▷ Candidate landmark is the centre of P
        break
      end if
    end for
  end for
  $\mathbf{s}_c \leftarrow$CONSTRAINSHAPE($\mathbf{s}$)
  for all points $\mathbf{x}_i$ in $\mathbf{s}$ **do**
    if $d(\mathbf{x}_i, \mathbf{x}_{c_i}) < \delta$ **then** ▷ Constrained point close to original?
      Update C_j with P_j
    else
      Discard P_j
    end if
  end for
end procedure

5.2 Discussion

The defining characteristic of SMAT is its incremental model. Allowing for the model to evolve as the object changes helps create a robust and highly specific model. On the downside, the absence of prior information makes the algorithm specially sensible to the first frames of the sequence. Still, this is an improvement over the previous works [Matthews 03][Kaneko 02] that relayed on the first frame alone.

Incorrect initialization in the first frames introduces undesirable exemplars in the model, creating clusters that would lead to further tracking of erroneous characteristics. This *model tainting* also happens to a lesser degree when tracking is lost: new clusters with exemplars from the background may be created. See for example figure 5.3. Each row of squares on the right hand side is the group of clusters that represents the texture of the patch around a landmark. For example, the first two rows model the appearance of the eye's pupil, and the third and fourth the corners of the mouth. Each small square shows the representative of that particular cluster. The most recently updated cluster is marked with a red dot. The tracking is lost due to total occlusion in figure 5.3(b), and

Algorithm 5.2 SMAT Constrain Shape

Let $U = \{\mathcal{U}_1, \ldots, \mathcal{U}_m\}$ be the set of n clusters that model the shape, with weights $\{w_1, \ldots, w_m\}$ and membership thresholds $\{T_1, \ldots, T_m\}$

procedure ConstrainShape(**s**)
  $\mathbf{s}' \leftarrow removeTranslation(\mathbf{s})$ ▷ Translate to origin
  for all cluster $\mathcal{U}_k$ in U **do**
    if $d_k(\mathbf{s}') < \tau_k$ **then** ▷ d_k as in equation 5.7
      Obtain $\mathbf{s}'_c$ with equation 5.9
      return
    end if
  end for
  Create a new cluster $\mathcal{U}_{m+1}$ from $\mathbf{s}'$
  Set $w_{m+1} \leftarrow 0$
  $U \leftarrow U \cup \mathcal{U}_{m+1}$
  if $m + 1 > K^S$ **then** ▷ Remove the cluster with lowest weight
    Find $\mathcal{U}_k \mid w_k \leq w_i \quad i = 1, \ldots, m$
    $U \leftarrow U \setminus \mathcal{U}_k$
  end if
end procedure

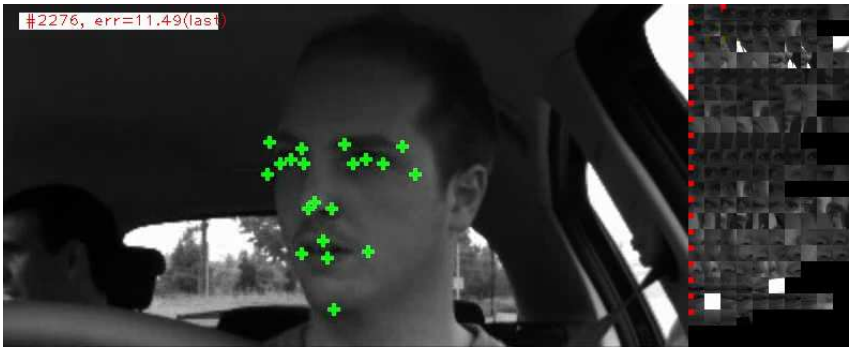
then correctly repositioned. However, some clusters in 5.3(c) now have a representative that is a bright white exemplar, which is not a possible texture of the corresponding patch.

SMAT keeps up to M exemplars per cluster, so the exemplar being used as representative can change as the model evolves. This feature may have, however, several drawbacks. With new exemplars modifying the clusters, they could overlap as they move in the feature space, wasting memory space and reducing the quantity of the space that can be modeled by the K clusters. In our experiments, we have observed that the opposite situation occurs much more frequently: the representative of the clusters rarely change after the cluster has reached a certain number of elements.

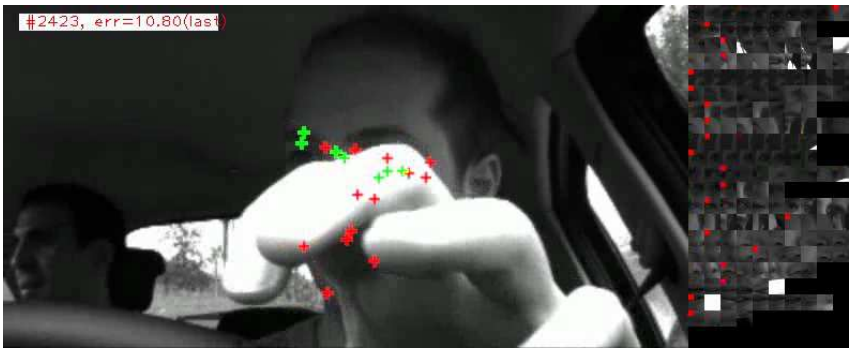
Enforcing a limit to the maximum number of clusters K makes clusters with low weight be discarded. Most of the discarded clusters will have a very low exemplar count, and many will be outliers. However, when all clusters have enough weight, the newest cluster will almost always have the lowest weight, and will be discarded if the incoming exemplar is not matched by any cluster. Under these circumstances, the model is not able to learn any new appearance, and effectively ceases to evolve. This is only noticeable on long video sequences, well over a thousand frames.

Other problem of the clustering used by SMAT is overfitting. When very similar patches are constantly introduced, one of them will be chosen as representative of one of the clusters, and as the size limit M is reached, exemplars further away will be discarded, reducing the variance of the cluster. This is specially clear when the membership in equation 5.1 is used. (This case may take place, for example, when the driver stays still and there are no illumination changes. At a frame rate of 25 or 30 fps, with M set to 50, the cluster will overfit in less than 2 seconds). This procedure will discard valuable information and future, subtle changes to the feature will lead to the creation of another cluster.

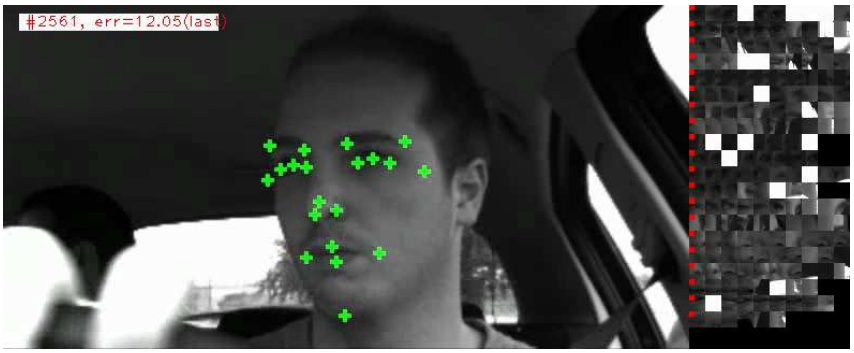
As implemented in SMAT, clustering is conceptually simple, and can be computed in real-time even when tracking several features simultaneously, and for reasonably large



(a) Before tracking loss



(b) Tracking loss



(c) Model fitted after tracking loss

Figure 5.3: An example of model tainting

patches. Many other classes of algorithms that can be computed incrementally could have been used. For example, Principal Components Analysis, used in ASMs and CLMs (see previous chapters), has been extended to support incremental updates of the eigenvectors and eigenvalues [Li 04, Artac 02]. Nonetheless, we have tried to address some of the shortcomings of the model building procedure of SMAT while keeping the model general structure, by proposing and testing different incremental clustering algorithms, presented below. With these alternative clustering methods, and alternative point distribution methods from section 5.4, we propose a variation of the original SMAT that we will call Robust SMAT (R-SMAT).

5.3 Alternative clustering methods

Incremental clustering deals with the problem of clustering a set of points in space that are presented sequentially, maintaining a set of clusters as optimal as possible in some sense. Most research in this field is related to document retrieval, database processing [Can 93] and data mining in general. Online databases where addition and removal of elements is frequent, or databases where due to the dimensionality of the points not all of them can be stored or accessed simultaneously are classic examples. A few approaches are mentioned here, we refer the reader to literature reviews on clustering for more details [Jain 99, Rasmussen 92]. Chapters 16 and 17 in [Manning 08] present clustering in the field of information retrieval.

Arguably the simplest and most frequently used incremental clustering method is the leader algorithm [Hartigan 75, Spath 80], detailed in algorithm 5.3. Each cluster $\mathcal{C}_i$ is defined by only one exemplar, and a fixed membership threshold T.

Algorithm 5.3 Leader clustering

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

procedure LEADER(E, C) ▷ cluster patch E
  for all $\mathcal{C}_i \in C$ **do**
    if $d(\mathcal{C}_k, E) < T$ **then** ▷ $E \in \mathcal{C}_k$
      UPDATEWEIGHTS($w_1^t, \ldots, w_n^t$) as in eq. 5.3
      return
    end if
  end for
  Create new cluster $\mathcal{C}_{n+1}$, with E as representative.
  Set $w_{n+1}^{t+1} \leftarrow 0$
  $C \leftarrow C \cup \mathcal{C}_{n+1}$
  if $n + 1 > K$ **then** ▷ Remove the cluster with lowest weight
    Find $\mathcal{C}_k \mid w_k \leq w_i \quad i = 1, \ldots, n$
    $C \leftarrow C \setminus \mathcal{C}_k$
  end if
end procedure

It starts by making the first exemplar the *representative* of a cluster. If an incoming exemplar fulfills a condition (usually being within a distance of the representative), it is marked as member of that cluster, otherwise it becomes a cluster on its own. The leader algorithm has been extended to work with fuzzy clusters [Asharaf 03], and interval data [Asharaf 06].

In the field of information retrieval, Hierarchical Agglomerative Clustering (HAC) [Willett 88] has been the *de-facto* strategy. Its basic working principle is to consider each point as a cluster, and merge these clusters until their number is as small as desired. HAC compute dendograms (hierarchy trees) based on some similarity measurement. The tree structure is fundamental as search refinement is equivalent to moving down the tree. In some applications, nodes down the structure are labeled with conceptual information, that may also be used in the clustering process [Fisher 87]. An in-depth review of the performance of some HAC methods can be found in [Charikar 97].

While HAC has some advantages, such as being deterministic and producing quality clustering, we have chosen to use methods that in most cases do not compute a hierarchy

and that may delete elements or clusters if needed. This choice is motivated by several reasons. On one hand, there is no labeling information available that could be computed to establish a hierarchy, and obtaining compact descriptions from the images poses an additional level of complexity. On the other hand, the gains in search speed obtained from using a tree structure are of little use in our application, as the aim is to produce a compact model to be used in matching against a wide range of patches, of which the best match will be then added to the model. A multi-scale approach would probably benefit from using HAC, but this is not our case.

The first clustering method tested in this chapter is the *leader* method. As this is probably the simplest clustering method, its performance can be a good baseline to which compare other approaches. The main benefits of the leader algorithm are its low complexity ($O(nk)$) and the low memory requirements ($O(k)$, where n is the dimension of the exemplars and k is the number of clusters) [Jain 99]. The membership threshold has been set as a fixed value (see section 5.5). As only the cluster representatives are stored, many more clusters can be kept in memory. In order to remove clusters generated by outliers, the maximum number of clusters has been limited.

Algorithm 5.4 LeaderP clustering

Let $C = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ be a set of n clusters, with weights $\{w_1^t, \ldots, w_n^t\}$ and membership thresholds $\{T_1, \ldots, T_n\}$

procedure LEADERP(E, C) ▷ cluster patch E
  for all $\mathcal{C}_i$ **do** ▷ $R_{\mathcal{C}_i}$ is the representative of $\mathcal{C}_i$
    if $d(R_{\mathcal{C}_k}, E) < T_i$ **then** ▷ $E \in \mathcal{C}_k$
      UPDATEWEIGHTS($w_1^t, \ldots, w_n^t$) as in eq. 5.3
      if C_i fixed **then**
        $d_j \leftarrow d(R_{\mathcal{C}_k}, E)$ ▷ $d_r,\ r = 0, \ldots, r-1$ contain past distances
        $T_k \leftarrow \tau(Sdv(d_1, \ldots, d_k))$
      else
        $R_{\mathcal{C}_k} \leftarrow median(\mathcal{C}_k)$
        if $|\mathcal{C}_k| = P$ **then**
          FIX($\mathcal{C}_k$) as in algorithm 5.5
        end if
      end if
      return
    end if
  end for
  Create new cluster $\mathcal{C}_{n+1}$, with E as representative.
  Set $w_{n+1}^{t+1} \leftarrow 0$
  $C \leftarrow C \cup \mathcal{C}_{n+1}$
  if $n + 1 > K$ **then** ▷ Remove the cluster with lowest weight
    Find $\mathcal{C}_k \mid w_k \leq w_i \quad i = 1, \ldots, n$
    $C \leftarrow C \setminus \mathcal{C}_k$
  end if
end procedure

It was referred above that the representative of the clusters in SMAT rarely changes after a few exemplars have been added. A modification of the leader algorithm has also been tested, where instead of making the first exemplar the representative and only

member of a newly created cluster, the first few exemplars added to the cluster are kept, up to P. Algorithm 5.4 describes the steps.

The median of the cluster is chosen as the representative, as in section 5.1.1. When the number of exemplars in the cluster reaches P, all exemplars but the representative are discarded, and it starts to work under the leader algorithm. P is chosen as a small number (we use 10). The membership threshold is however flexible: the distances between the representative and each of the exemplars that are found to be members of the cluster is saved, and the variance of those distances is used to calculate the threshold. Because distance is a scalar, many values can be kept in memory without having a impact on the overall performance. Note that this is not possible in the original SMAT method, as the values become invalid when the representative changes. We will refer to this method as *leaderP*.

Algorithm 5.5 LeaderP clustering (cont)

```
26: procedure FIX(C_k)                          ▷ Fix the representative of the cluster
27:     R_{C_k} ← median(C_k)
28:     r ← 0
29:     for all E_j ∈ C_k, E_j ≠ R_{C_k} do
30:         d_r ← d(E_j, R_{C_k})
31:         r ← r + 1
32:     end for
33:     Keep R_{C_k}, discard the rest of the elements
34:     T_k ← τ(Sdv(d_1, ..., d_r))
35: end procedure
```

Finally, a hierarchical method is proposed. Exemplars are added to the clusters as in the original clustering in SMAT, and are sub-clustered again using the *leaderP* algorithm, with a smaller value of P. This method is summarized in algorithm 5.6.

Using a hierarchy helps reduce the memory requirements and the number of distances to be computed, while the risk of overfitting is diminished. If many, very similar patches were included, they would be grouped in one of the sub-clusters, and the other sub-clusters would remain unmodified. The representative of the cluster can shift from sub-cluster to sub-cluster. This modification also includes the possibility of merging clusters and sub-clusters. When the maximum number of clusters K is reached, the distance between the cluster with lowest weight $\mathcal{C}_l$ and the rest is measured. If for a cluster $\mathcal{C}_t$ the distance $d_{tl} = d(\mathcal{C}_t, \mathcal{C}_l)$ is below a threshold T^1_{merge}, the clusters are merged. If the distance d_{tl} is below T^2_{merge}, with $T^2_{merge} > T^1_{merge}$, the distance between the representative of $\mathcal{C}_t$ and the sub-clusters in $\mathcal{C}_l$ is computed, and sub-clusters closer than T^1_{merge} are added to $\mathcal{C}_t$. This process is summarized in algorithm 5.7 and depicted in figure 5.4, where just one of the subclusters ($\pentagon$) of cluster $\mathcal{C}_3$ is made part of $\mathcal{C}_1$, and the rest is discarded. The representative of $\mathcal{C}_1$ and the membership threshold change as a result.

This hierarchical structure is similar to the Leader-Subleader algorithm in [Vijaya 04]. Our method, however, builds the clusters and sub-clusters simultaneously, and does not require several passes over the exemplar set.

This alternative clustering methods will be tested in section 5.5 along the original method in [Dowson 05]. The one that performs best will be chosen to be part of R-SMAT.

Algorithm 5.6 Hierarchical clustering

Let $C = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ be a set of n clusters, with weights $\{w_1^t, \ldots, w_n^t\}$ and membership thresholds $\{T_1, \ldots, T_n\}$
Let $S^j = \{\mathcal{S}_1^j, \ldots, \mathcal{S}_m^j\}$ be the set of subclusters of a cluster $\mathcal{C}_j$

procedure HIERARCHICAL(E, C) ▷ cluster patch E
  for all $\mathcal{C}_i$ **do** ▷ $R_{\mathcal{C}_i}$ is the representative of $\mathcal{C}_i$
    if $d(R_{\mathcal{C}_k}, E) < T_i$ **then** ▷ $E \in \mathcal{C}_k$
      UPDATEWEIGHTS($w_1^t, \ldots, w_n^t$) as in eq. 5.3
      LEADERP(E, S^k)
      $R_{\mathcal{C}_k} \leftarrow median(\mathcal{C}_k)$
      T_k
    end if
    return
  end for
  Create new cluster $\mathcal{C}_{n+1}$, with E as representative.
  Set $w_{n+1}^{t+1} \leftarrow 0$
  if $n + 1 > K$ **then** ▷ Remove the cluster with lowest weight
    Find $\mathcal{C}_l \mid w_l \leq w_i \quad i = 1, \ldots, n$
    MERGECLUSTERS($C, \mathcal{C}_l$)
    $C \leftarrow C \setminus \mathcal{C}_l$
  end if
end procedure

5.4 Alternative point distribution modeling methods

As for the appearance modeling, other methods instead of the original can be used for modeling the distribution of the patches.

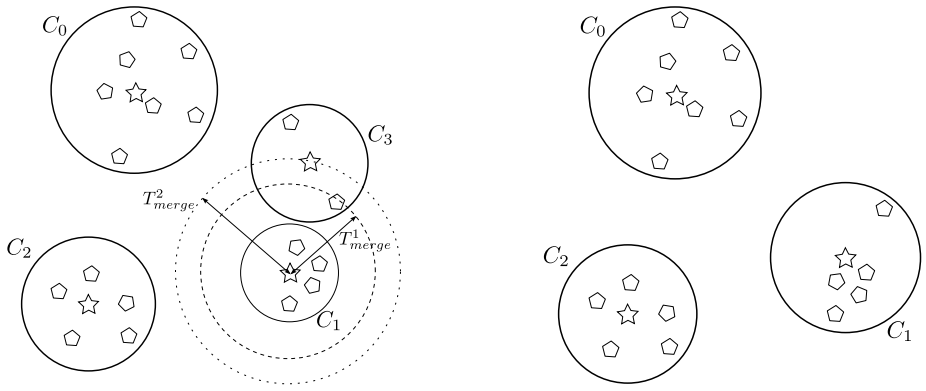
The constraints imposed on the point distribution could be of any kind, and the proposals for shape modeling in CLM and ASM could be used here, from splines to finite elements [Cootes 95]. In [Jimenez 09], a rigid 3D point model was used to track a face with a stereo camera configuration. SMAT was used to track 2D patches in the projection of the vertices of the model in each camera, effectively enforcing the perspective relations on the position of the patches.

The restrictions on using a pre-learned model for shape are less than those for an appearance model, as it is of lower dimensionality and the deformations are easier to model. It has been shown [Gross 05b] that location and tracking errors are mainly due to appearance, and that a generic shape model for faces is easier to construct. In chapter 4, the shape was modeled from the training samples using PCA. The same method has been used to model the deformation of the shape in SMAT. Note that both methods are basically the same, CLM computing the orthogonal vector base from hand-marked samples and SMAT from some of the shape seen in the current video sequence. We have tested both approaches, using *a priori* and on-line modeling.

Robust methods of obtaining the shape parameters $\mathbf{p}$ were also tested. In addition to the L_2 norm as in 4.2, M-estimators [Huber 81] were used. Let $\mathbf{s}$ be a given shape, and $\mathbf{s}_0$ and $(\mathbf{s}_1, \ldots, \mathbf{s}_N)$ be the mean shape and a series of orthonormal vectors defining deformations of the mean shape. Using L_2 norm, finding $\mathbf{p}$ is equivalent to minimize

Algorithm 5.7 Hierarchical clustering

```
1: procedure MergeClusters(C, C_l)
2:     for all cluster C_i ∈ C, C_i ≠ C_l do
3:         if d(C_i, C_l) < T^1_merge then
4:             C_i ← C_i ∪ C_l                          ▷ Merge all subclusters
5:             return
6:         end if
7:     end for
8:     for all cluster C_i ∈ C, C_i ≠ C_l do
9:         if d(C_i, C_l) < T^2_merge then
10:            for all subclusters S^j ∈ C_l do
11:                if d(C_i, S^j) < T^1_merge then
12:                    C_i ← C_i ∪ S^j                  ▷ Merge subcluster
13:                end if
14:            end for
15:        end if
16:    end for
17: end procedure
```



(a) $\mathcal{C}_3$ selected for removal

(b) Subcluster from $\mathcal{C}_3$ is now included in $\mathcal{C}_1$

Figure 5.4: Addition of clusters and subclusters

$$arg\min_{\mathbf{p}} ||\mathbf{s} - (\mathbf{s}_0 + \sum_{j=1}^{m} p_j \mathbf{s}_j)||^2 \tag{5.10}$$

The estimation of $\mathbf{p}$ using the L_2 norm is very sensitive to the presence of outlier points: a high error value from one point will severely influence the value of $\mathbf{p}$.

Robust methods allow for the rejection of outlier points in the estimation, by introducing a weight that depends on the magnitude of the residue of each point, measured from the current estimation. This minimization is a case of re-weighted least squared. The weight decreases more rapidly than the square of the residue, and thus a point with error tending to infinite will have zero weight in the estimation.

The function to minimize is

$$arg\min_{\mathbf{p}} \sum_{i=1}^{2n} \rho(r^{i^2}) \tag{5.11}$$

where r^i is the residue for coordinate i of the shape

$$r^i = \mathbf{x}^i - (\mathbf{s}_o^i + \sum_{j=1}^{m} p_j \mathbf{s}_j^i)) \tag{5.12}$$

where $\mathbf{x}^i$ are the points of the shape $\mathbf{s}$, and $\mathbf{s}_j^i$ is the ith element of the vector $\mathbf{s}_j$, and $\mathbf{s}_0$ is the base shape.

Several robust estimators have been tested: *Huber*, *Cauchy*, *Gaussian* and *Tukey* functions [Zhang 97]. Each robust function has a different performance depending on the distribution of outliers. A study was made in [Bergasa 06] that resulted in similar performance for all of them in a similar scenario to that of this thesis. The most frequently used is the Huber function, defined as

$$\rho(r) = \begin{cases} r^2/2 & \text{if} \quad |r| \leq \sigma \\ \sigma(|r| - \sigma/2) & \text{if} \quad |r| > \sigma \end{cases} \tag{5.13}$$

The derivative of the Huber function (its *influence function*) ψ and its weight function w are

$$\psi(r) = \frac{\partial \rho(r)}{\partial r} = \begin{cases} r & \text{if} \quad |r| \leq \sigma \\ \sigma(sign(r)) & \text{if} \quad |r| > \sigma \end{cases} \tag{5.14}$$

and

$$w(r) = \frac{\psi(r)}{r} = \begin{cases} 1 & \text{if} \quad |r| \leq \sigma \\ \sigma(|r|) & \text{if} \quad |r| > \sigma \end{cases} \tag{5.15}$$

We estimate the value of the scale parameter σ as a function of the median of the values of r. We have used the Huber function to test to what extent the result would improve with a robust fitting function. In the same way than the clustering techniques, the best point modeling method will be used in R-SMAT.

5.5 Tests and Results

Different configurations of the algorithm were tested using the video sequences in the RS-DMV database, and error values were obtained by comparing with the hand-marked positions.

Initialization of the positions is a particular problem for SMAT, even more than for CLM as the first frames of the sequence are key to building a good model. We propose to use STASM is used to initialize SMAT in the first frame. We consider the one-time delay of 0.4 seconds that STASM takes (on average) to process a frame acceptable. STASM, however, introduces an estimation error that influences the accuracy of SMAT for the rest of the sequence. A slightly incorrect initialization will make SMAT track the (slightly) erroneous points. To decouple this error from the evaluation of accuracy of SMAT in the tests, the SMAT's shape was initialized in the first frame with positions from the ground-truth data. For the models that used an *a priori* shape model, the vector of initial positions $\mathbf{x}_{t=0}$ was projected on the model, so the actual starting positions $\mathbf{x}'_{t=0}$ are

slightly different from the given ones. Actually the root mean square error that comes from the projection on the vector base is around a value of 2 pixels, depending on the sequence. In a real application scenario, where no ground-truth data is available, STASM would be used to initialize SMAT. The performance of the best configuration of SMAT with automatic initialization from STASM is evaluated at the end of this section.

Images from the video sequences are resized in the same way as in chapter 4. Keeping the aspect ratio, they are scaled down so their size is at most 320×240 pixels.

Starting from frame #0, the SMAT model tracks the feature points in the incoming frames. However, there are situations where the tracking is lost or fails for most points, resulting in a high error value. The same loss-detection measures implemented for CLM in chapter 4 were used for SMAT. The maximum rotation angles and the number of points allowed outside the Viola & Jones box are higher than in CLM, because SMAT proved to have better recuperation capabilities than CLM and was able to recover from moderate tracking losses by itself. For models with a trained shape model, when a tracking loss was detected the shape was repositioned to the center of the box found with Viola&Jones algorithm, the scale set to the mean of the previous 100 frames and the angle set to zero. The rest of the shape parameters were all set to zero. This method is very simple and may not be valid for other applications, but it is very easy to implement and has proved to serve its purpose for the videos in the database. This re-initialization is less precise than using STASM, but we found that the range of convergence for SMAT is big enough, and this initialization is of course much faster than running STASM. For the *on-line* shape model, STASM was used to reposition the landmarks.

To assess the performance of the algorithm and obtain results that would help us choose the best options to make part of R-SMAT, several model configurations were tested on each sequence. The configuration options include:

- Shape model: *a priori* or *on-line*.
- Robust shape parameter estimation: Huber function, as well as non-robust L_2 norm for the *a priori* models, and Mahalanobis distance for the *on-line* model.
- Patch size: 11×11 pixels, 15×15 pixels, and 20×20 pixels.
- Clustering method: *leader*, *leaderP*, *original* and *hierarchical*.

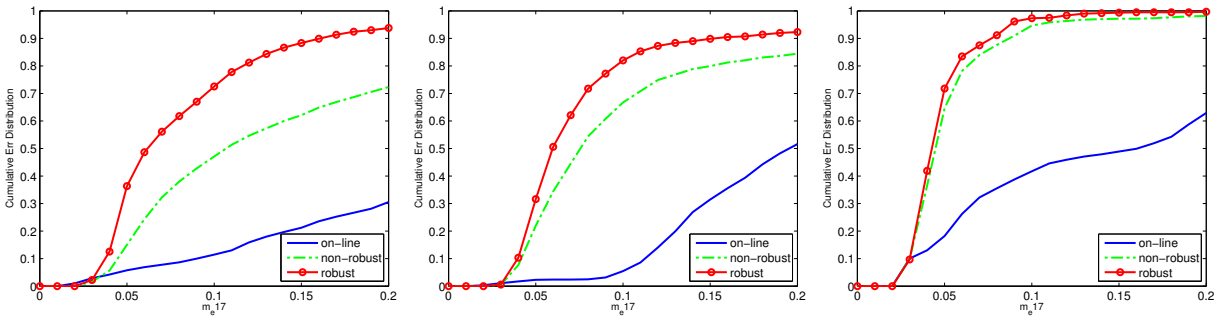
The different patch sizes would give an idea of the surface around each point that appropriately represents the feature being tracked. Smaller patches have bigger risk of drifting in a quick displacement, while bigger patches are more subject to noise and carry a performance penalty.

All possible combinations of the options above yield 36 different configurations to be run on the videos of the database. Reproducing graphs for all configurations on all types of videos would take too much space (108 graphs), and thus some options are dropped after a few comparisons, when it has been clearly demonstrated that there are better options. First, the shape model and parameter estimation options are tested. With the option that yields best results, tests are carried out on the different patch sizes, and finally results for the four clustering algorithms are presented.

5.5.1 Performance of different shape models and parameter estimation

Three shape model and parameter estimation options are tested: *on-line* model, *a priori* model with L_2 estimation and *a priori* model with robust estimation, using the Huber

function. Figure 5.5 presents a comparison of the results of the different shape models, and specific error plots for each shape model are shown in 5.6. In the latter figure, the fitting error without considering tracking losses in *keyframes* is plotted as a dotted red line. The other parameters of SMAT are fixed: all use patches of 15×15 pixels, with *leaderP* as the clustering method. Similar results are obtained for other patch sizes and clustering algorithms.



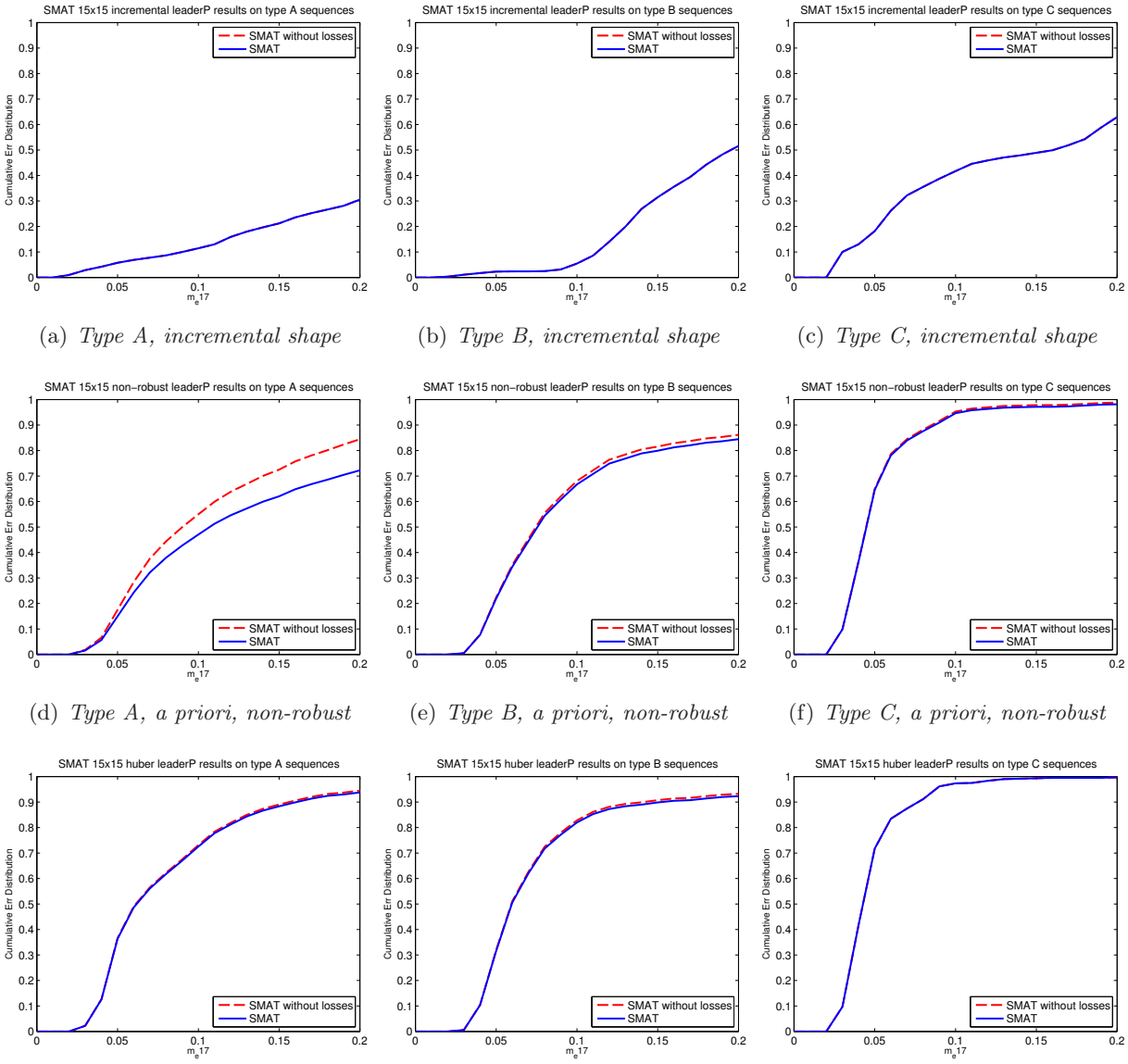
(a) *Type A* (b) *Type B* (c) *Type C*

Figure 5.5: Comparison of the performance of different shape models, with *leaderP* clustering

The results in figures 5.5 and 5.6 show that the *on-line* shape model performs poorly, followed by the trained model with non-robust fitting. Robust fitting obtains the best results, and it is the by far the best choice of all three. Robust shape estimation shows a reduced number of tracking losses, which translates to less model tainting. The reasons for the failure of the *on-line* shape model come from problems on its definition. One key step of the training of *a priori* models is removing the scale, rotation and translation from the shapes (using Procrustes or other method). In equation 5.6 only the translation is removed. This fact implies the introduction of exemplars in the clusters that are actually the same shape, with small differences on scale and rotation. On one hand, this limits the ability of the clusters to represent actual shapes. On the other, the projection of a shape on the eigenvectors of the cluster in equation 5.7 can change considerably if it is rotated, leading to an improper reconstruction when equation 5.9 is enforced. It should be noted that fewer points than the 20 points in our shape model were used in previous works using SMAT. In [Dowson 05], tests included 4 or 5 points, and in [Dowson 06], most examples used just one element per tier. While the *on-line* shape model may work with a small number of points, it does not work with a model involving 20 points.

		Mean	Maximum	Minimum
on-line	*Type A*	0%	0%	0%
	Type B	0%	0%	0%
	Type C	0%	0%	0%
non-robust	*Type A*	14.15%	20.65%(seq. #6)	5.04%(seq. #4)
	Type B	3.45%	3.93%(seq. #9)	0.16%(seq. #10)
	Type C	0.30%	0%(seq. #12,#13)	1.21%(seq. #11)
robust	*Type A*	0.99%	2.08%(seq. #6)	0%(seq. #1,#2,#6)
	Type B	1.13%	1.96%(seq. #9)	0%(seq. #10)
	Type C	0%	0%	0%

Table 5.1: SMAT track losses for different shape models



(a) *Type A, incremental shape* (b) *Type B, incremental shape* (c) *Type C, incremental shape*

(d) *Type A, a priori, non-robust* (e) *Type B, a priori, non-robust* (f) *Type C, a priori, non-robust*

(g) *Type A, a priori, Huber* (h) *Type B, a priori, Huber* (i) *Type C, a priori, Huber*

Figure 5.6: Cumulative error distribution of SMAT using 3 different shape models, all with *leaderP* clustering and 15×15 patches

Table 5.1 shows the track losses for the different types of sequences, as a percentage of the *keyframes* in those sequences. The *on-line* shape model has some apparently surprising values, with no losses for all types of sequences. The fact is, tracking is not working for some periods those videos, but instead of drifting, the landmarks of the shape get stuck for tens of frames. As the head of the subjects does not displaces much horizontally, the tracking loss detection does not identify a failure for any of those frames. At the same time, the fitting error is high, as demonstrated by figures 5.6(a), 5.6(b) and 5.6(c). A few frames of the *type A* sequence #1 fitted with the *on-line* shape model can be seen in figure 5.7. Model points are marked with a green cross, and a red cross indicates that the point has displaced very far from the previous position, and it is considered lost, or that a new cluster has been created for that landmark in that frame. Please note how in frames 30 and 80 (5.7(d) and 5.7(e)), one of the points displaces to the upper left corner and

stays there, because no constraints are applied on the distribution of the points. When a tracking loss is detected, the Viola & Jones box is drawn in that frame, and a message in a red box is printed on screen. This message is printed for 20 frames, for easier visualization: at 30 frames per second, it could otherwise go unnoticed. Figure 5.7(m) presents the m_e17 error for the sequence. Although losses appear in the images and are detected, they do not took place on *keyframes*, and thus do not reflect on table 5.1 and the error plot.

The *a priori* shape model obtains better results, but it is sensitive to fitting errors and outliers disrupt the estimation of the model parameters. Figures 5.8 and 5.9 show a few frames of the *type A* sequence #1 and *type C* sequence #11, respectively. In the former, the model drifts remarkably under small occlusions from the driver's hand. Head turns, even when not pronounced, also lead to tracking losses. For the second sequence, the driver does not move much. Because the texture of the hand is somewhat similar to the face skin, the occluded points do not drift much, and the shape stays in place. Still, a random error takes place around frame 1135 (5.9(e)), and the model is close to lose track. The model recovers by itself, but the error in those frames is high. Tracking losses detected on *keyframes* are marked with a red asterisk in the m_e17 error plots. In figure 5.9, other samples from *type C* sequence #11 show again the effect of a small occlusion in the model fitting.

Robust fitting of the *a priori* shape model shows a clear advantage in both fitting accuracy and tracking losses, as demonstrated by the error graphs and table 5.1. Figures 5.10 and 5.11 show the results of SMAT using a robust shape model in the same sequences as the figures above. Note that the Y axis scale is 10 times smaller than in figure 5.7(m) and 3 times smaller than in 5.8(m). The influence of the occlusions and head rotations is much smaller than in the other cases above. Points marked in red denote that this point is considered to be an outlier, and their actual position has been corrected to one that belongs to the subspace of shapes that the vector base represents.

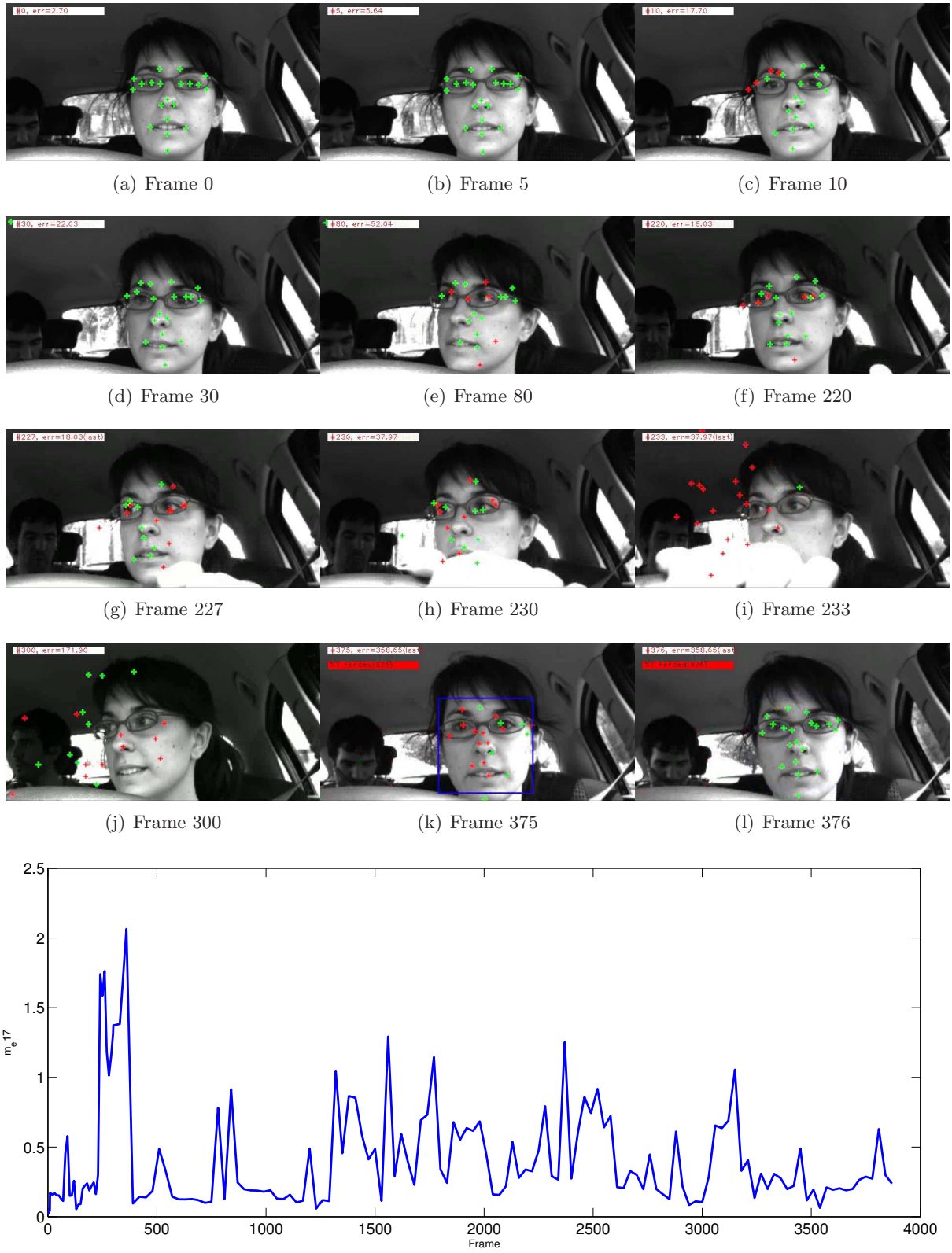
All tests described from now on use robust fitting of the shape model with the Huber function, which is selected for R-SMAT.

5.5.2 Performance of models with different patch sizes

The same patch sizes used with CLM were tested. Robust fitting was used in all tests, and *leaderP* as the clustering method. Figure 5.12 shows the results of the SMAT using the three different sizes, all with *leaderP* clustering. The graphs correspond to all *keyframes*, including frames were the tracking is lost. All three patch sizes show similar results, which are nearly identical for sequences of *type B* and *type C*. Sizes of 15×15 and 20×20 perform better than 11×11 for *type A* sequences. The difference comes from the amount of lost frames, which is clear in figure 5.13(a), where the distance between the overall graph and the graph not including tracking losses stands out. One of the main characteristics of *type A* sequences is frequent head movements, and when those happen motion blur appears. Small features are more easily lost, and thus working with a larger patch improves the chances of success.

Table 5.2 summarizes the percentage of losses in *keyframes* for the 3 patch sizes. Tests with 11×11 patches have the highest number of tracking losses of all three sizes.

Results using the other clustering algorithm were produced, in order to check that the results are independent of the clustering method used. Figure 5.12 presents the results for the original SMAT clustering algorithm. The plots show that while performance defers, the comparison of the results based on patch sizes and the conclusions drawn hold. Sizes 15×15 and 20×20 perform similarly, while 11×11 performs a little worse, specially



(a) Frame 0 (b) Frame 5 (c) Frame 10

(d) Frame 30 (e) Frame 80 (f) Frame 220

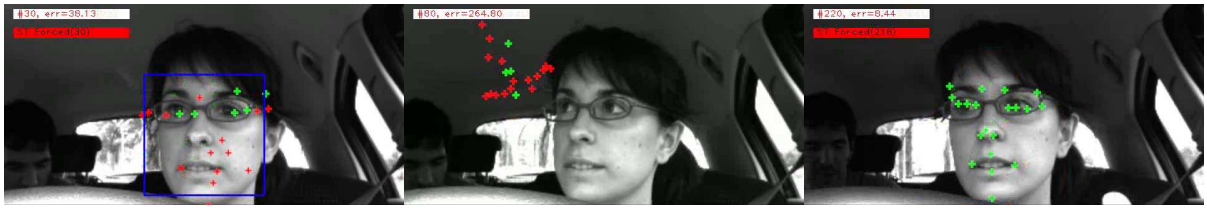
(g) Frame 227 (h) Frame 230 (i) Frame 233

(j) Frame 300 (k) Frame 375 (l) Frame 376

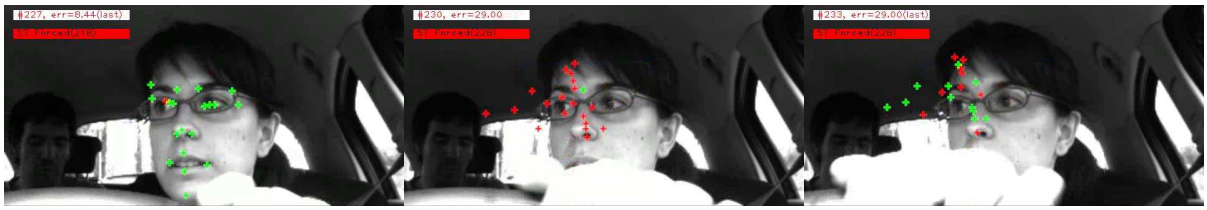
(m) m_e17 error for the sequence

Figure 5.7: Samples of *type A* sequence #1, fitted with an on-line shape model

(a) Frame 0 (b) Frame 5 (c) Frame 10

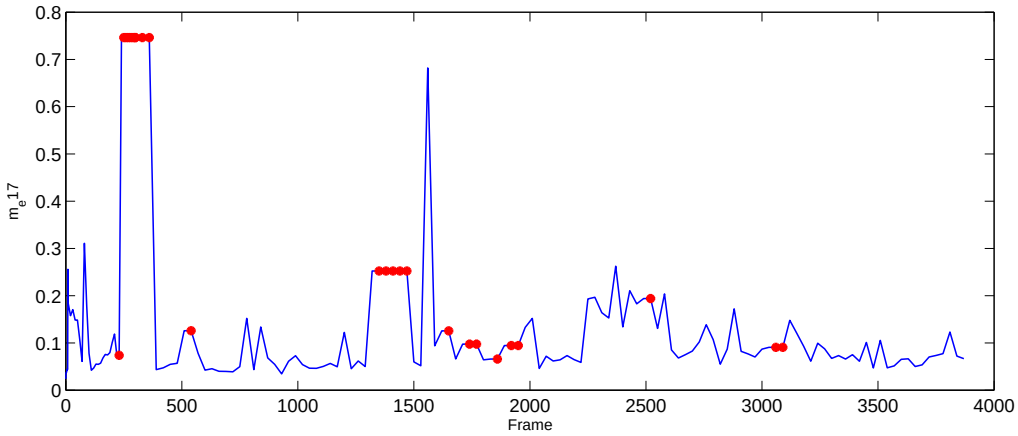


(d) Frame 30 (e) Frame 80 (f) Frame 220



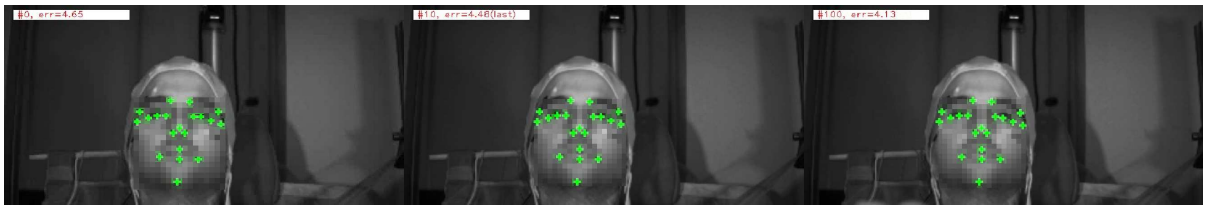
(g) Frame 227 (h) Frame 230 (i) Frame 233

(j) Frame 300 (k) Frame 375 (l) Frame 376

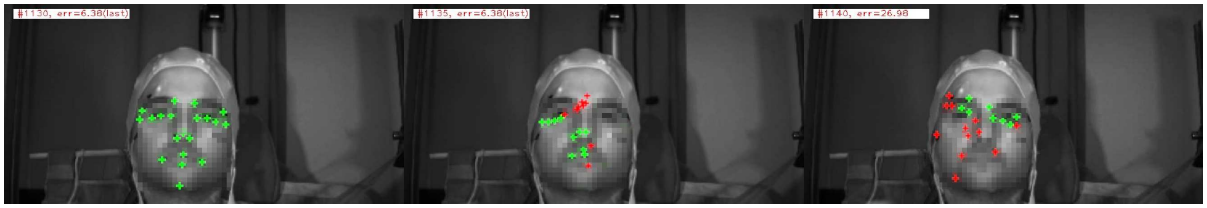


(m) m_e17 error for the sequence

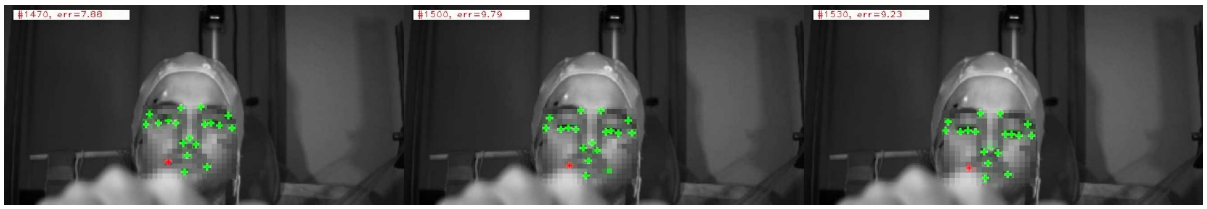
Figure 5.8: Samples of *type A* sequence #1, fitted with an *a priori* shape model with non-robust fitting



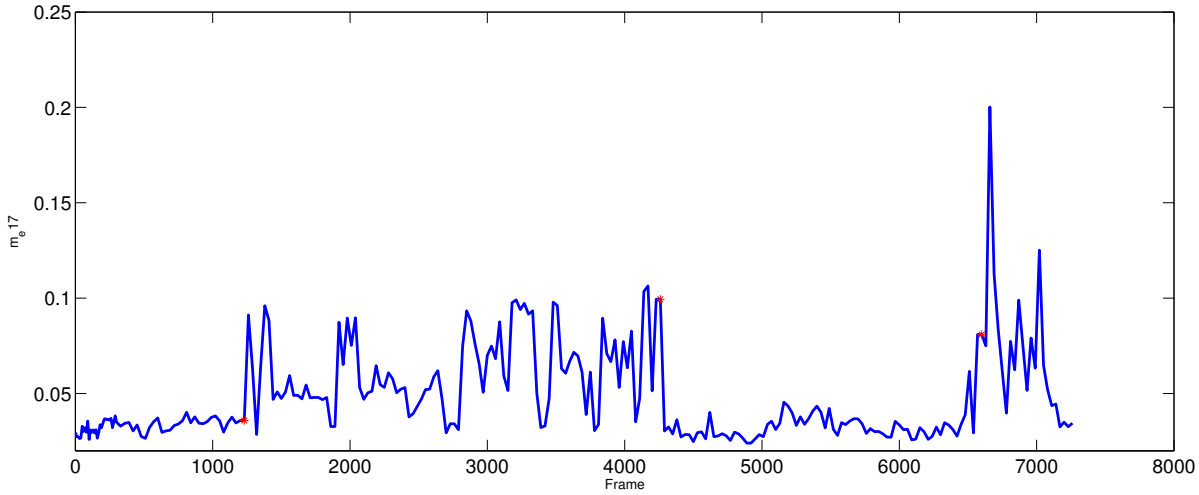
(a) Frame 0 (b) Frame 10 (c) Frame 100



(d) Frame 1130 (e) Frame 1135 (f) Frame 1140



(g) Frame 1470 (h) Frame 1500 (i) Frame 1530



(j) m_e17 error for the sequence

Figure 5.9: Samples of *type C* sequence #11, fitted with an *a priori* shape model with non-robust fitting. Pixelated for privacy

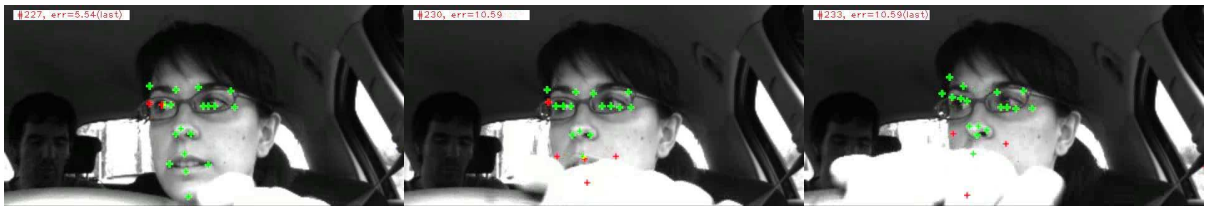
for *type A* sequences. Considering the similar performance of the former sizes, patches of 15×15 pixels are selected for R-SMAT, and used for the rest of the chapter. A model with patches of 20×20 would deliver similar performance.

5.5.3 Performance of clustering algorithms

Finally, the four clustering algorithms' performance is compared. Figure 5.15 shows the m_e17 error of SMAT on the 3 types of sequences. All models use the robust shape fitting

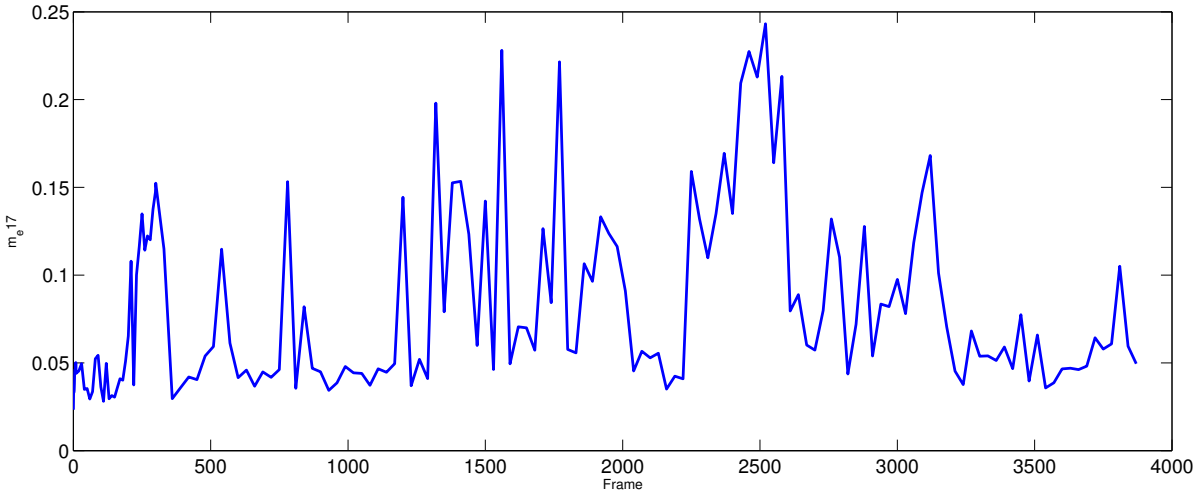
(a) Frame 0 (b) Frame 5 (c) Frame 10

(d) Frame 30 (e) Frame 80 (f) Frame 220



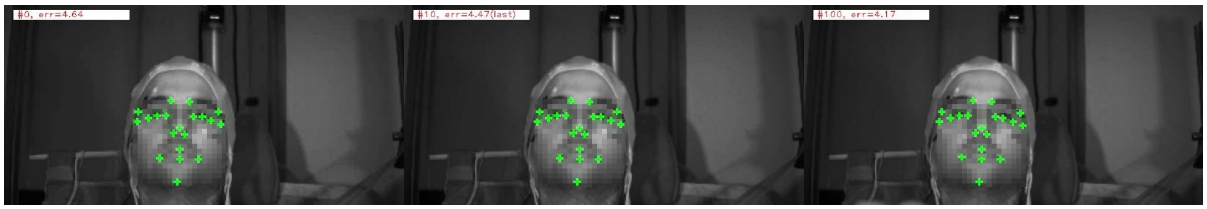
(g) Frame 227 (h) Frame 230 (i) Frame 233

(j) Frame 300 (k) Frame 375 (l) Frame 376

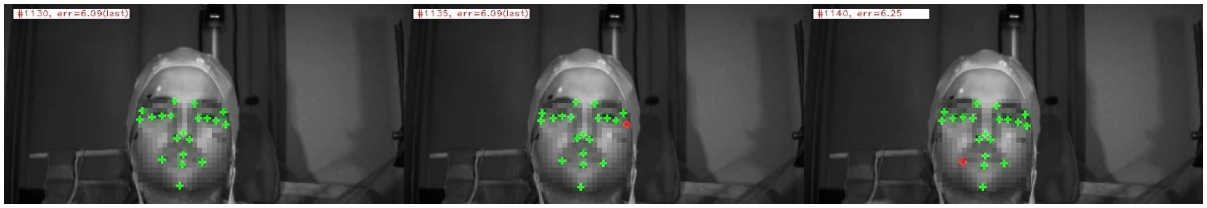


(m) m_e17 error for the sequence

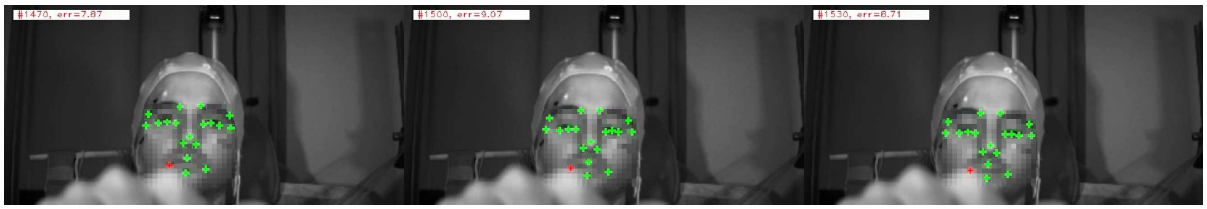
Figure 5.10: Samples of *type A* sequence #1, fitted with an *a priori* shape model with robust fitting



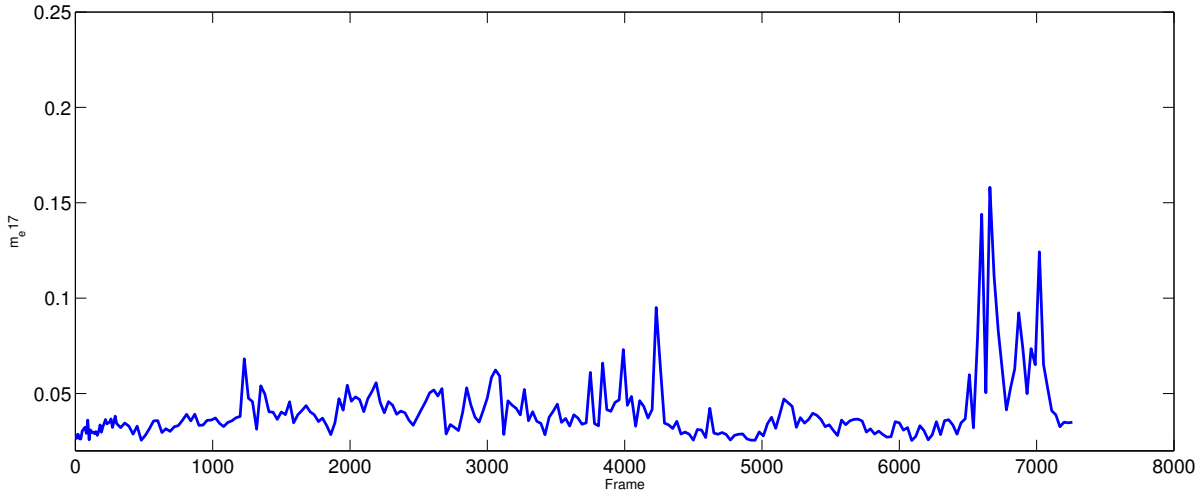
(a) Frame 0 (b) Frame 10 (c) Frame 100



(d) Frame 1130 (e) Frame 1135 (f) Frame 1140



(g) Frame 1470 (h) Frame 1500 (i) Frame 1530

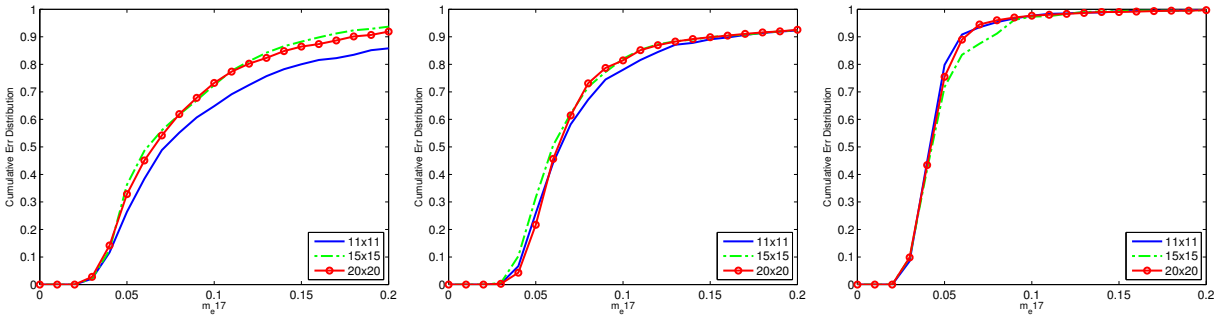


(j) m_e17 error for the sequence

Figure 5.11: Samples of *type C* sequence #11, fitted with an *a priori* shape model with robust fitting. Pixelated for privacy

method (Huber) and patches of 15×15.

From the graphs, it is clear that the original SMAT clustering is the worst performer. This is specially evident in 5.15(b). We stated in 5.2 that the original clustering method could lead to overfitting, and *type B* sequences are specially prone to this: patches are usually dark and do not change much from frame to frame, and the subject does not move frequently. When a movement takes place, it leads to high error values, because the model has problems finding the features. Table 5.3 shows the tracking losses of each



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

Figure 5.12: Comparison of the performance of different patch sizes, with *leaderP* clustering

		Mean	Maximum	Minimum
11×11	*Type A*	5.87%	12.5% (seq. #5)	3.47%(seq. #7)
	Type B	0.76%	1.14%(seq. #9)	0%(seq. #10)
	Type C	0%	0%	0%
15×15	*Type A*	0.99%	2.08%(seq. #7)	0%(seq. #1,#2,#6)
	Type B	1.13%	1.96%(seq. #9)	0%(seq. #10)
	Type C	0%	0%	0%
20×20	*Type A*	0.33%	1.43%(seq. #4)	0%(seq. #1,#2,#6 #7)
	Type B	1.52%	4.58%(seq. #9)	0%(seq. #10)
	Type C	0%	0%	0%

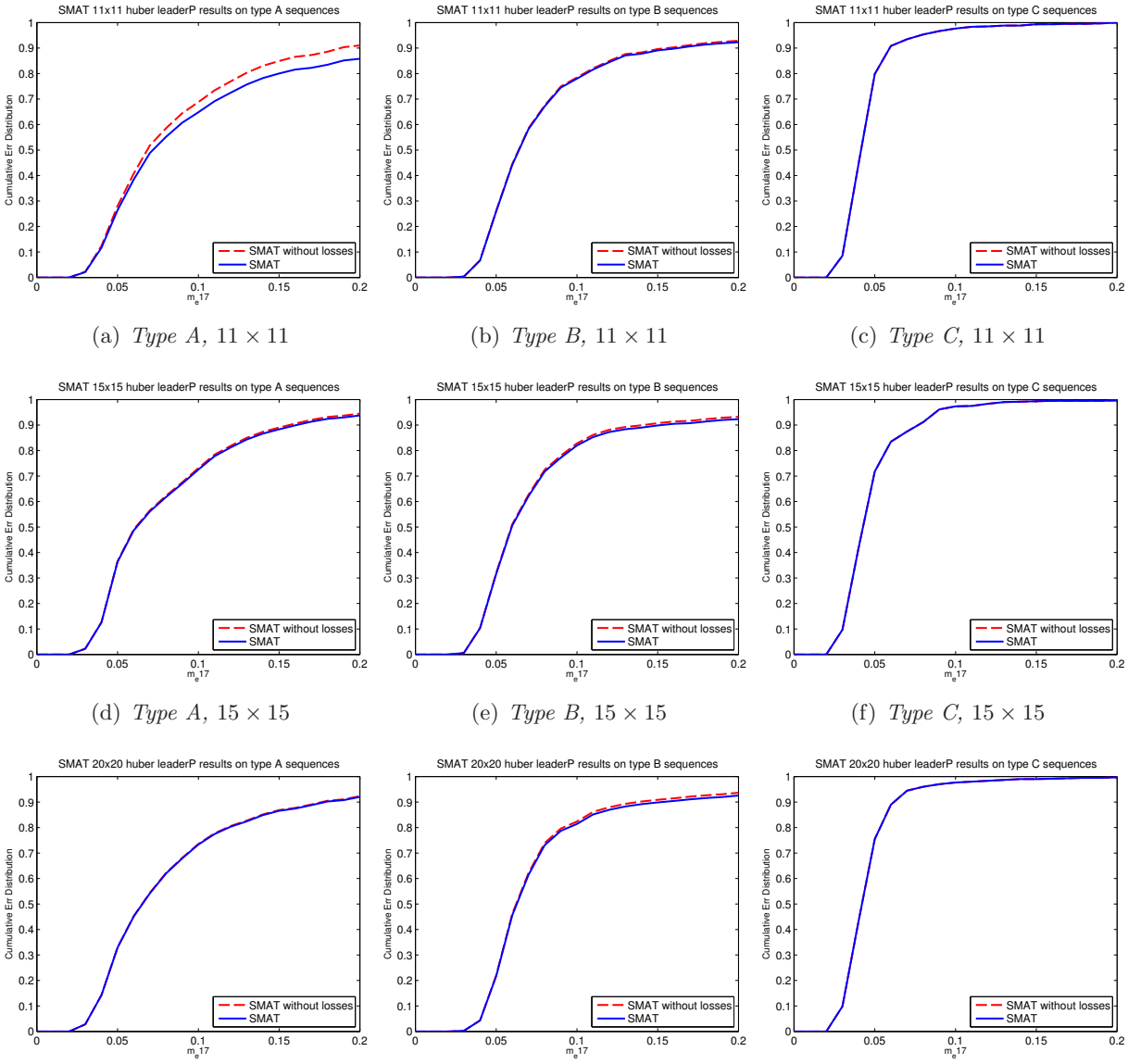
Table 5.2: SMAT track losses for different patch sizes, with *leaderP* clustering

clustering method, as a percentage of the *keyframes* in the sequences. Only the *leader* and to a least extent the *hierarchical* method have a remarkable number of losses for any kind of sequences. Most of the losses come for just one sequence, #7 and #6 for *leader* and *hierarchical* respectively. Figure 5.16 shows the corresponding plots for each type of sequence and clustering.

		Mean	Maximum	Minimum
leader	*Type A*	5.21%	18.05% (seq. #7)	0%(seq. #1,#2)
	Type B	0.81%	2.12%(seq. #9)	0%(seq. #10)
	Type C	0%	0%	0%
leaderP	*Type A*	0.99%	2.08%(seq. #7)	0%(seq. #1,#2,#6)
	Type B	0.71%	1.96%(seq. #9)	0%(seq. #10)
	Type C	0%	0%	0%
original	*Type A*	1.77%	5.03%(seq. #4)	0%(seq. #1,#2,#5)
	Type B	1.03%	2.45%(seq. #9)	0%(seq. #10)
	Type C	0%	0%	0%
hierarchical	*Type A*	3.65%	10.6%(seq. #6)	0%(seq. #1,#2)
	Type B	0.32%	0.82%(seq. #9)	0%(seq. #10)
	Type C	0%	0%	0%

Table 5.3: SMAT track losses for different clustering methods

The other 3 clustering methods, *leader*, *leaderP* and *hierarchical* perform similarly,



(a) *Type A*, 11×11 (b) *Type B*, 11×11 (c) *Type C*, 11×11

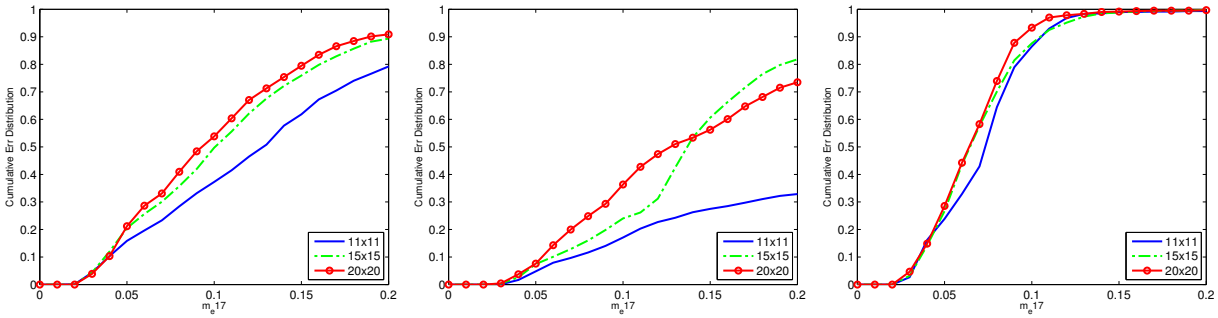
(d) *Type A*, 15×15 (e) *Type B*, 15×15 (f) *Type C*, 15×15

(g) *Type A*, 20×20 (h) *Type B*, 20×20 (i) *Type C*, 20×20

Figure 5.13: SMAT cumulative error distribution for 3 different patch sizes. All tests use *leaderP* clustering and robust shape fitting

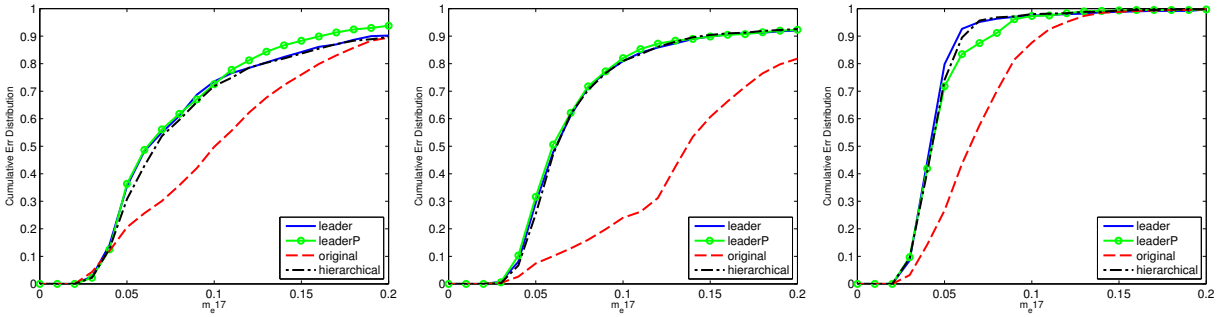
with *leaderP* showing slightly better results for *type A* and *type B*, and slightly worse for *type C*. That such a simple clustering as *leader* can perform as good as more complicated methods indicates that the texture of the patches can be modeled with a small number of exemplars. Other scenarios, with stronger deformations or changes in appearance may require more complicated clustering, and methods like *leaderP* and *hierarchical* would then outperform *leader*.

In addition to the influence of the clustering method, the patch size and the robust fitting method, there is a limit in the accuracy of the SMAT model imposed by the shape model. Only a limited space of deformations can be synthesized by the shape model, so a certain amount of error appears if the ground-truth values do not belong to that shape space. This is specially the case of head turns, as the shape model was trained on frontal faces, and only the robust fitting makes SMAT keep track of the visible points on



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

Figure 5.14: Comparison of the performance of different patch sizes, with the SMAT original clustering



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

Figure 5.15: Comparison of the performance of different clustering algorithms

self-occluded faces.

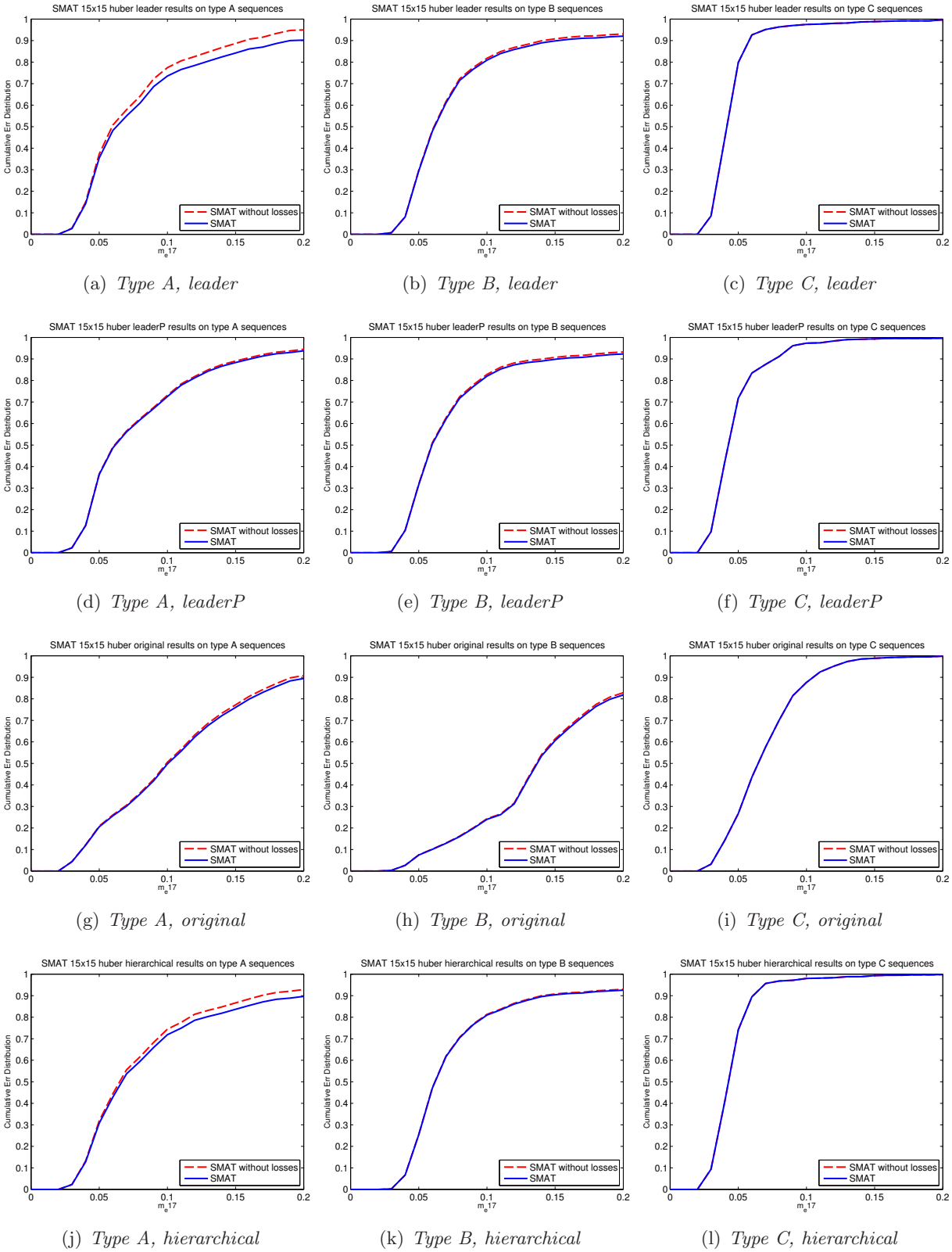
The best combination of options for the algorithm has been shown to be a robust fitting function, patches of 15×15 pixels and any of the 3 alternative clustering methods. Modified to use this configuration, SMAT improves in robustness and accuracy. We call this new method Robust SMAT, or *R-SMAT*.

5.5.4 Initializing R-SMAT with STASM

Results presented so far have been obtained initializing SMAT and R-SMAT with landmarks from the handmarked ground-truth data. In a real scenario, an automatic algorithm would be used to initialize SMAT and R-SMAT. An evaluation of the fitting error of R-SMAT when initialized automatically is presented here.

We have used STASM for this task. STASM was run on the first frame of each sequence, and its estimation seeded the position of R-SMAT in that video. R-SMAT would track the points given by STASM from then on, so any error in the initialization will remain for the rest of the sequence, although many other variables, such as the similitude between the patches around the STASM landmarks and around the ground-truth data will influence the final fitting error. Figure 5.17 plots the error distributions of R-SMAT when initialized with STASM. Errors for the same model with initialization from ground-truth data are also plotted for reference.

As expected, the figure shows the results worsen for all types of sequences. But the



(a) *Type A, leader* (b) *Type B, leader* (c) *Type C, leader*

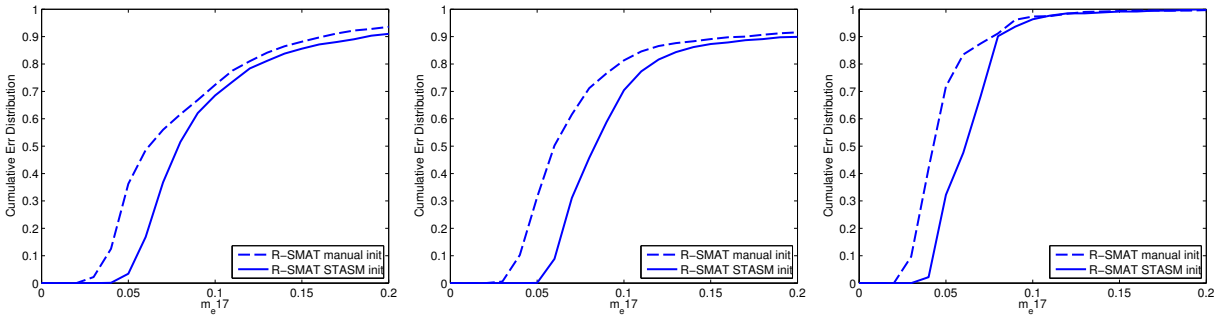
(d) *Type A, leaderP* (e) *Type B, leaderP* (f) *Type C, leaderP*

(g) *Type A, original* (h) *Type B, original* (i) *Type C, original*

(j) *Type A, hierarchical* (k) *Type B, hierarchical* (l) *Type C, hierarchical*

Figure 5.16: SMAT cumulative error distribution for different clustering algorithms

lost accuracy is relatively small, with a 5% loss at $m_e17 = 0.1$ for *type A* sequences, 10% loss for *type B* and just 1% for *type C*. The mean of the m_e17 error of STASM in the first



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

Figure 5.17: R-SMAT cumulative error distribution for manual and automatic initialization

frame is 0.0571 for *type A* sequences, 0.0805 for *type B* sequences and 0.0514 for *type C*. The higher error for *type B* may explain part of the lost accuracy in figure 5.17(b).

5.5.5 Processing times

One of the most important requirements for R-SMAT is for it to run in real-time. Table 5.4 summarizes the average execution speed for R-SMAT in seconds and frames per second, for some representative configurations. The worst frame processing times are close to the limit, but these are extreme cases that occur infrequently. For all frames, figure 5.18 plots the average processing time of the previous 30 frames ($\sim$ 1 second). The graph has strong similarities to figure 5.10(m), where the error values for that sequence with R-SMAT can be found. More challenging frames, with occlusions or head turns, involve longer searches and more computation, and this reflects in processing times. For any 30-frame window, the average is far from the 30 fps (0.033 seconds per frame) limit, drawn as a dotted line in the upper part of the figure.

Configuration	Mean (fps)	Sdv (fps)	Worst frame (fps)
leader clustering, 20×20	89.39	31.65	31.05
leaderP clustering, 15×15	112.56	32.80	36.86
leaderP clustering, 20×20	78.09	30.77	32.78
hierarchical clustering, 15×15	102.60	24.15	48.57

Table 5.4: Execution time for several configurations of R-SMAT in frames per second

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

5.5.6 Comparison of R-SMAT vs. STASM

This section closes with a comparison between our proposal R-SMAT and STASM. In chapter 4 it was demonstrated that the extensions over the original ASM that STASM includes make it very accurate in all sequences, and able to work reliably under occlusions,

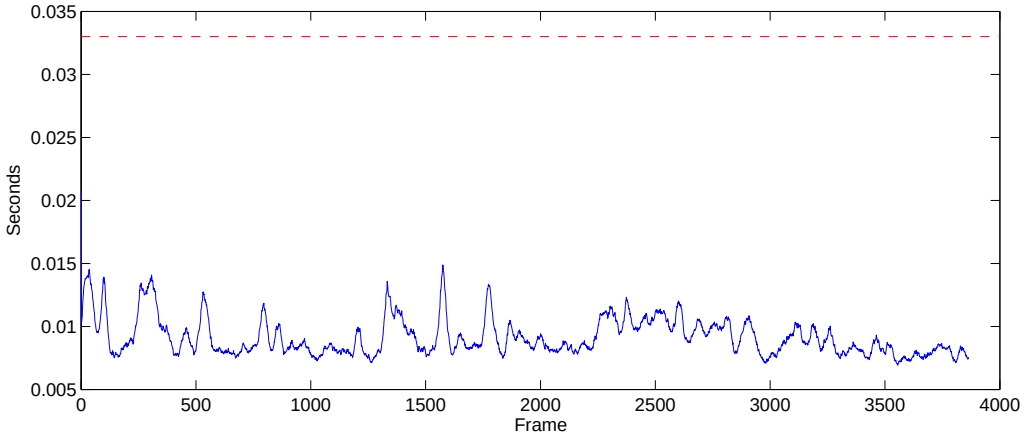
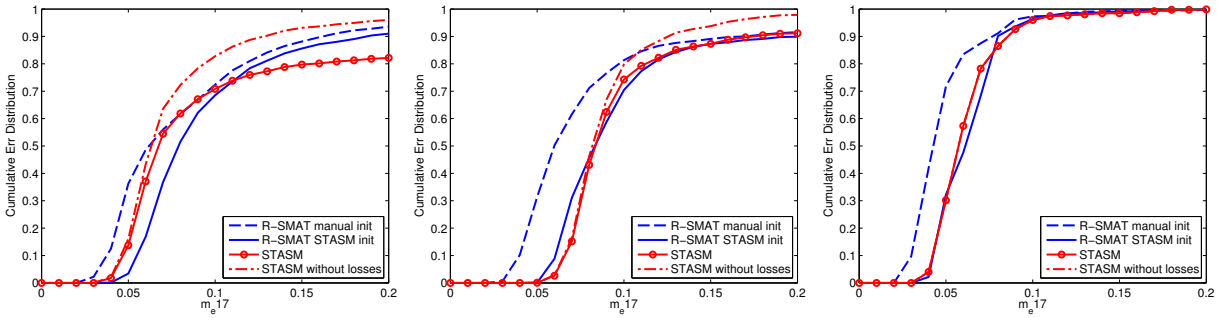


Figure 5.18: Average processing time of R-SMAT of the previous 30 frames, for *type A* sequence # 1

although it was not able to detect the face when the driver turned his/her head. Another problem, critical to our application, is that it does not work in real time.

R-SMAT is able to work in real time, but its main drawback is that it requires a few frames to initialize its appearance model properly. For this matter, it is desirable that the drivers stays facing frontal to the cameras for a few instants during initialization. If the model is not built correctly in those frames, the tracking will not succeed. Using a pre-learned shape model and a robust function for shape fitting, it has shown to be robust to head turns and self-occlusions. Because the texture model updates constantly, it is able to work under different illumination conditions. Texture is modeled using small patches around the shape landmarks, which help make it robust to lighting changes.



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

Figure 5.19: Comparison of the performance of STASM and SMAT

Figure 5.19 shows the m_e17 error plots of both models. R-SMAT graph is shown for both initializations, automatic and from ground-truth (manual), and STASM is plotted with and without considering losses. For all types of sequences, R-SMAT initialized manually outperforms STASM when losses are considered. Expectedly, STASM shows better accuracy than R-SMAT when tracking is not lost (i.e., when the face is frontal). The R-SMAT plot is the first to rise from the X axis, and is not crossed by the STASM without losses until approximately $m_e17 = 0.1$ in 5.19(a), and later for the other figures. R-SMAT initialized with STASM performs almost identically as STASM in 5.19(b) and

5.19(c), and slightly worse for *type A* sequences.

The introduction of this thesis stated a series of requirements that a proper system our the application must fulfill. We briefly compare the performance of R-SMAT and STASM on extracts of sequences from RS-DMV that reflected these requirements.

Head turns Figures 5.20 show a few frames of both models fitted to the face of the driver in sequence #5. Error is plotted in figure 5.21. Losses are indicated as a small dot in the STASM graph. R-SMAT did not get lost in this sequence. The small images on top of the plot show the fitting of R-SMAT for those particular frames. The driver rotates his face in various moments, and STASM can not fit in some of them (frames 60, 150, 170 and 280).

Facial gestures and talking Figures 5.20 and 5.22 present several frames with the drivers engaged in a conversation. Other figures presented below contain gestures and drivers actively talking. Both R-SMAT and STASM handle facial gestures and talking without problems.

Occlusions Frames from sequence #7 are shown in figure 5.23. In addition to head turns and talking, the lower part of the face of the driver is often occluded by his hand. Figure 5.25 presents the error plots for both methods. The partial occlusion around frame 2600 does not translate into an increase in the fitting error for neither STASM nor R-SMAT.

A total occlusion takes place around frame 2400, shown in figure 5.24, where the hand of the driver completely occludes the face. This occlusion leads to a tracking loss, depicted as a very high error in figure 5.25. This figure only shows the results for R-SMAT, STASM did not locate the face in any of the frames.

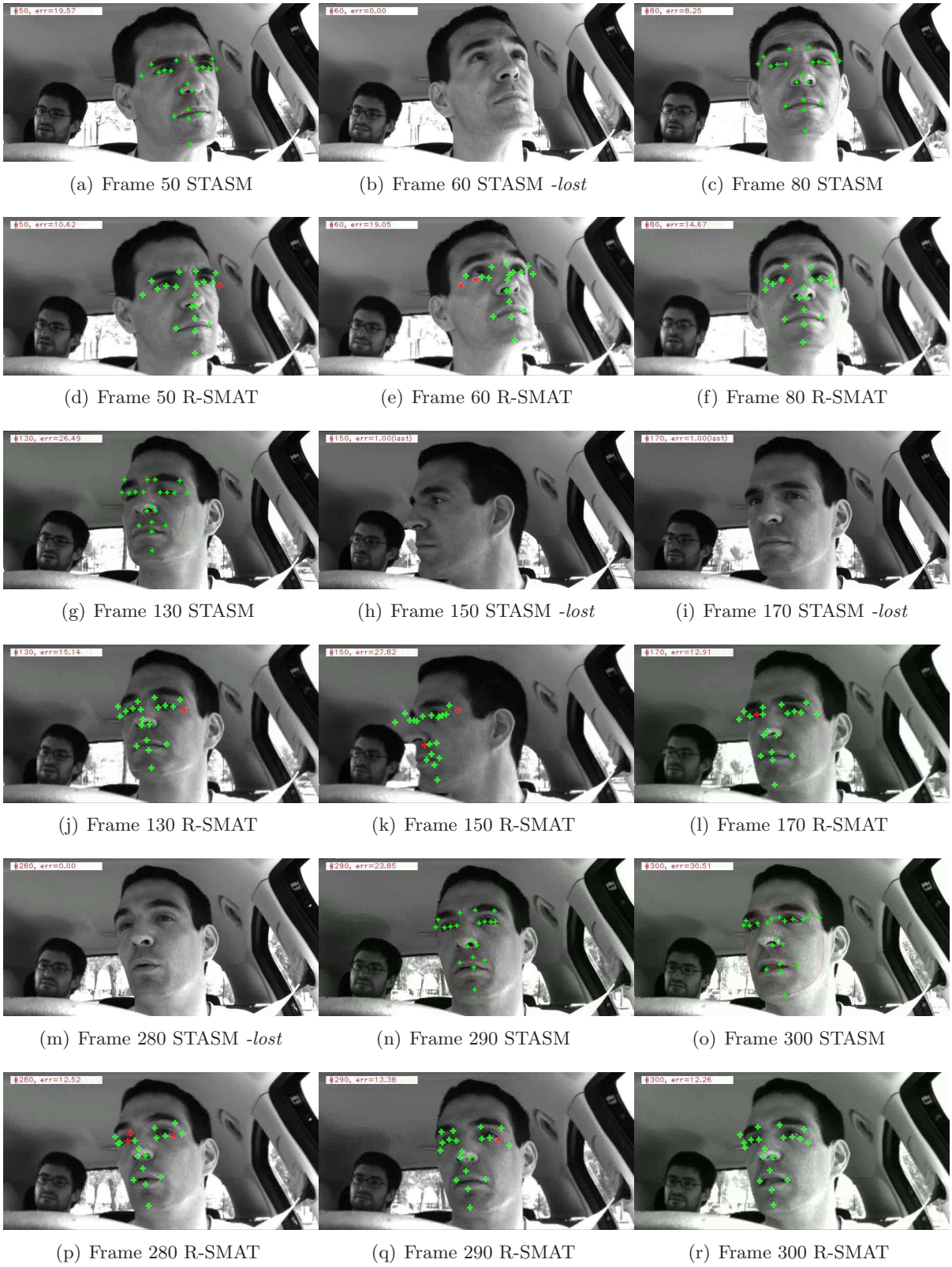
Drivers wearing glasses Sequences #1 and #2 feature the same driver with and without glasses. R-SMAT and STASM perform well in both cases, indicating an independence from the presence or absence of glasses. Figure 5.26 shows a few frames from both sequences, with R-SMAT and STASM fitted.

Illumination changes and nighttime operation Quick illumination changes take place in the sequences recorded outdoors, as the car passes through the shadows of trees and buildings. Figure 5.27 shows a few frames from sequence #6 that are an example of these situations. STASM and R-SMAT fit to the face correctly, and with low error. The error plot is shown in figure 5.28.

RS-DMV videos recorded in simulators, specially *type B* are very dark and close to a real nighttime environment. Figure 5.29 shows a few samples from sequence #9, with R-SMAT and STASM fitted. The driver in the images talks and gestures frequently. Both methods perform well, with almost no losses and good accuracy, as presented in see table 5.3 and figure 5.16.

5.6 Conclusions and contributions

This chapter has presented the Simultaneous Modeling and Tracking (SMAT) method, as introduced by [Dowson 05]. Instead of using a model built *a priori* like Constrained Local Models (CLM) or Active Shape Models (ASM), SMAT builds a model of texture



(a) Frame 50 STASM (b) Frame 60 STASM *-lost* (c) Frame 80 STASM

(d) Frame 50 R-SMAT (e) Frame 60 R-SMAT (f) Frame 80 R-SMAT

(g) Frame 130 STASM (h) Frame 150 STASM *-lost* (i) Frame 170 STASM *-lost*

(j) Frame 130 R-SMAT (k) Frame 150 R-SMAT (l) Frame 170 R-SMAT

(m) Frame 280 STASM *-lost* (n) Frame 290 STASM (o) Frame 300 STASM

(p) Frame 280 R-SMAT (q) Frame 290 R-SMAT (r) Frame 300 R-SMAT

Figure 5.20: Head rotation: STASM and R-SMAT fitted sequence #5

and a point distribution *on-line*, as new frames become available in the video sequence. Texture in the patch around each landmark is represented by a group of clusters, built

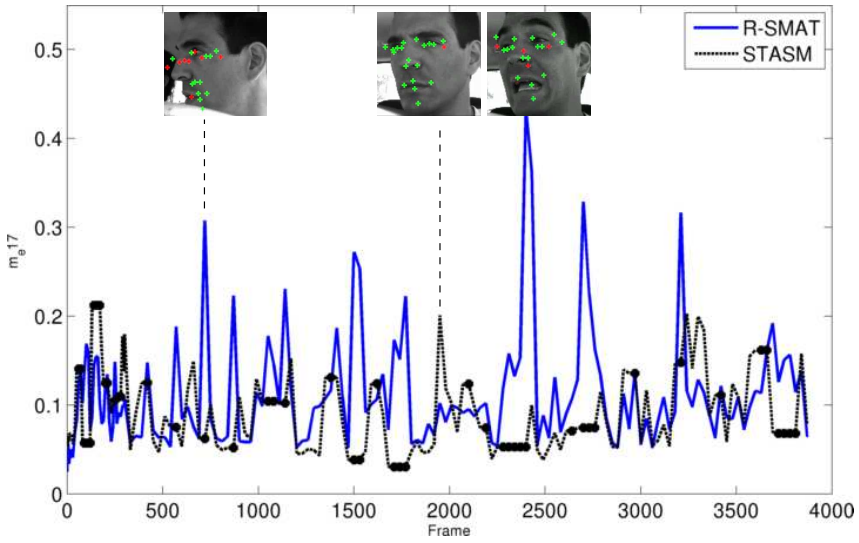
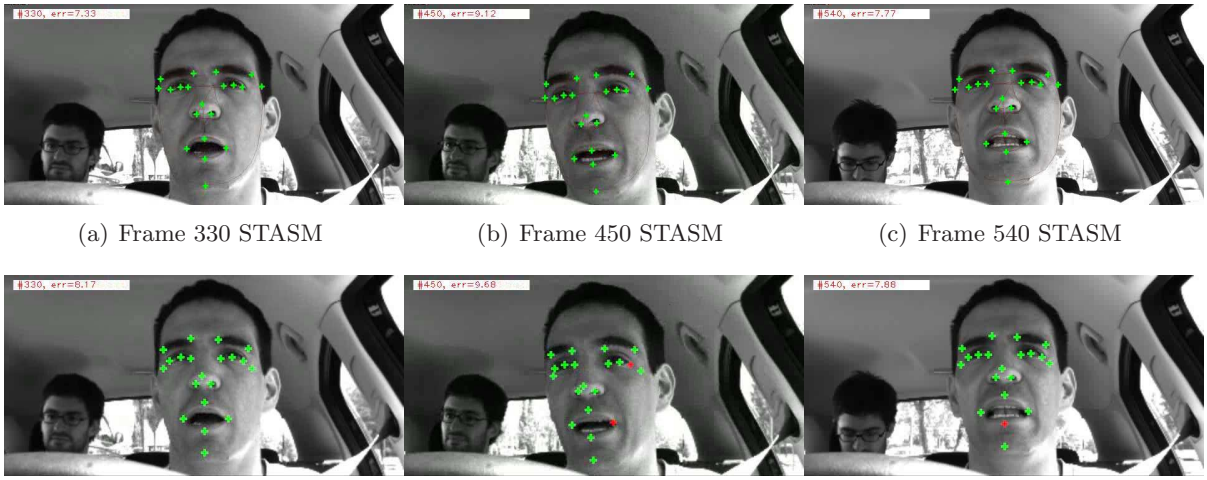


Figure 5.21: Error plots for STASM and R-SMAT in sequence # 5



(a) Frame 330 STASM (b) Frame 450 STASM (c) Frame 540 STASM

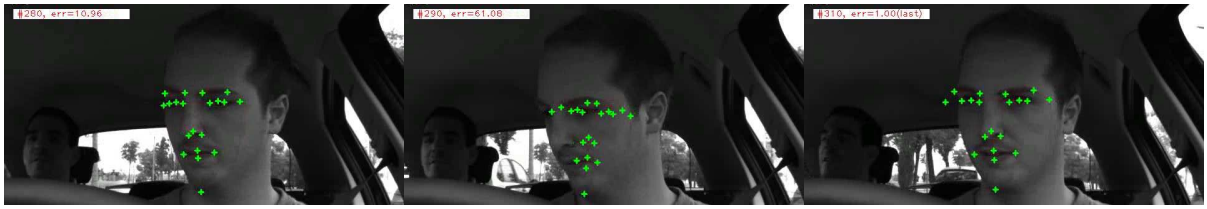
(d) Frame 330 R-SMAT (e) Frame 450 R-SMAT (f) Frame 540 R-SMAT

Figure 5.22: Driver talking: STASM and R-SMAT fitted in sequence #5

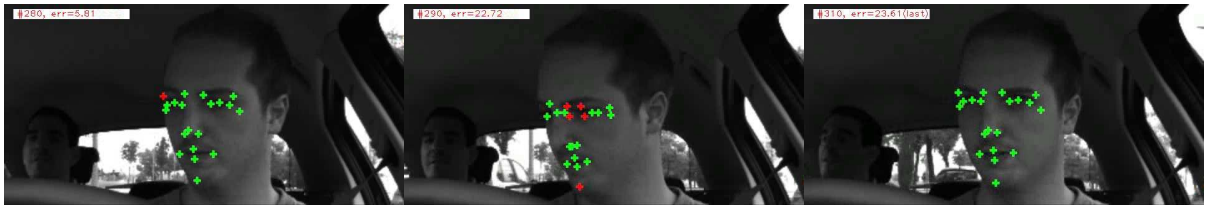
incrementally. In every frame, texture is extracted and added to the model. The same process applies to the shape.

Several problems have been identified in the original proposal. The clustering method is prone to overfitting in some situations, and may saturate, discarding new exemplars and thus loosing its ability to model upcoming variations of texture or shape. Three alternative clustering methods have been proposed to create a texture model with better characteristics. Shape modeling with clusters was found to be more difficult, and an point distribution model as used by CLM was proposed as a solution. Constraining the position of the landmarks with the model was done with both a L_2 norm and the Huber function, a robust M-estimation.

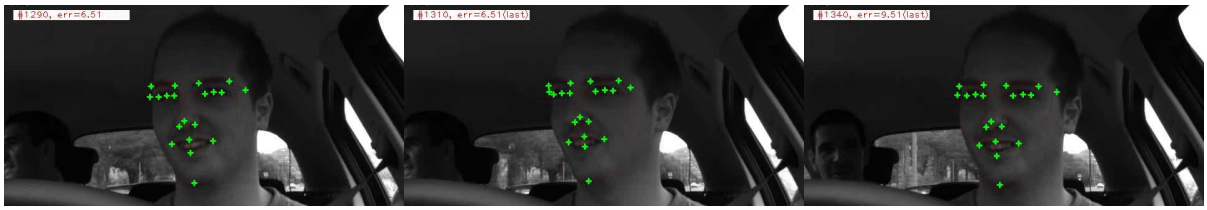
Test have been carried out on the three shape models, and the three new and the original clustering methods, using the sequences in the database. In addition, three different patch sizes (11×11, 15×15 and 20×20) were tested. The results clearly show that the *a priori* shape model with robust fitting obtains lower error and improved robustness



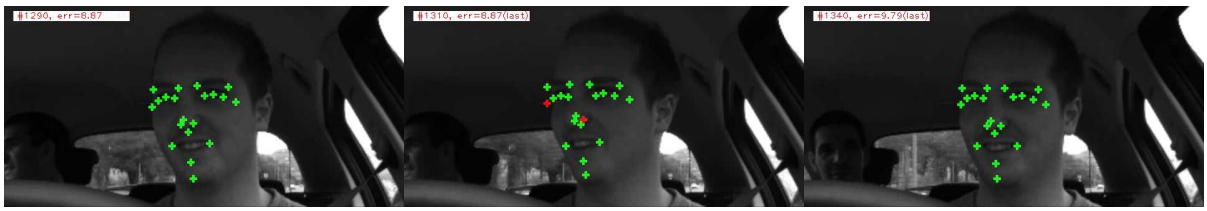
(a) Frame 280 STASM (b) Frame 290 STASM (c) Frame 310 STASM



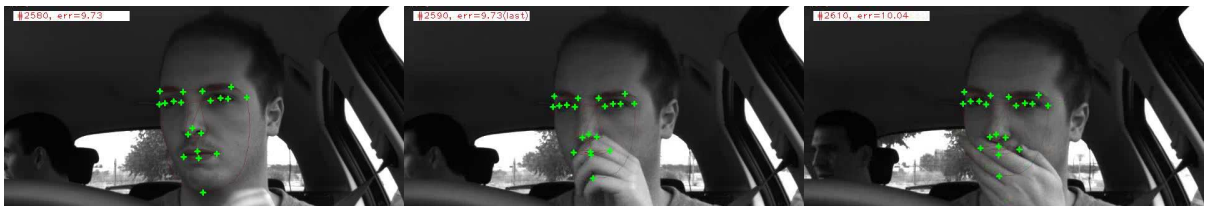
(d) Frame 280 R-SMAT (e) Frame 290 R-SMAT (f) Frame 310 R-SMAT



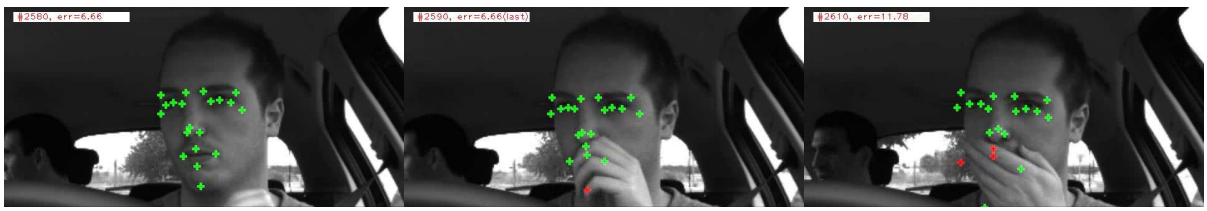
(g) Frame 1290 STASM (h) Frame 1310 STASM (i) Frame 1340 STASM



(j) Frame 1290 R-SMAT (k) Frame 1310 R-SMAT (l) Frame 1340 R-SMAT



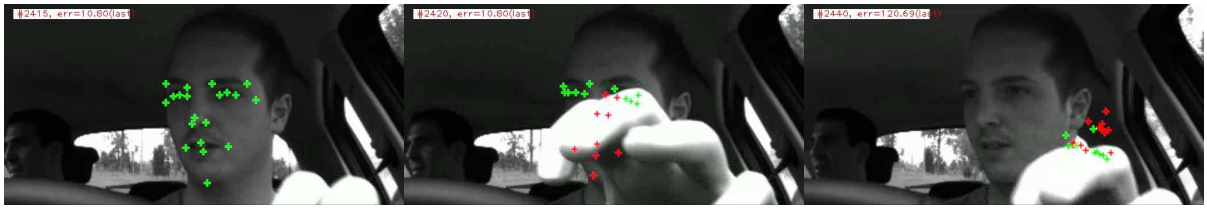
(m) Frame 2580 STASM (n) Frame 2590 STASM (o) Frame 2610 STASM



(p) Frame 2580 R-SMAT (q) Frame 2590 R-SMAT (r) Frame 2610 R-SMAT

Figure 5.23: Occlusions and head turns: STASM and R-SMAT fitted in sequence #7

to occlusions and head rotations. Any of the new clustering methods outperforms the original one by a wide margin. Performance does not show a strong dependency on the



(a) Frame 2415 R-SMAT (b) Frame 2420 R-SMAT (c) Frame 2440 R-SMAT

Figure 5.24: Total occlusion of the face in sequence #7

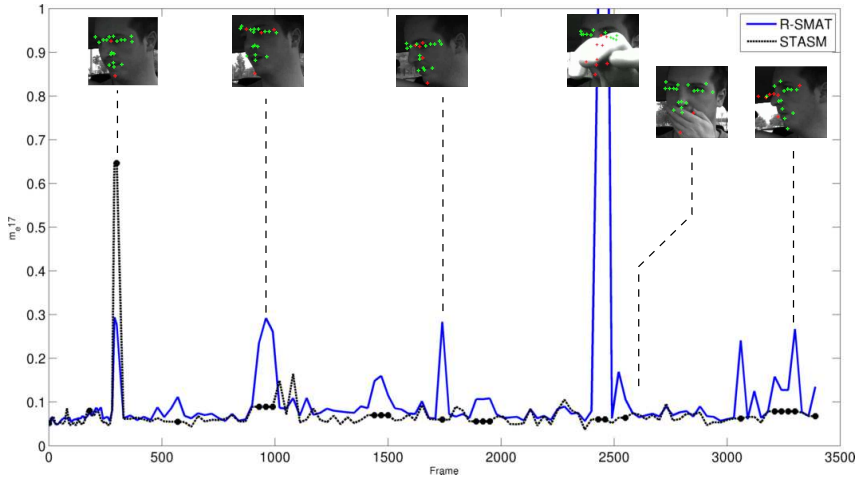


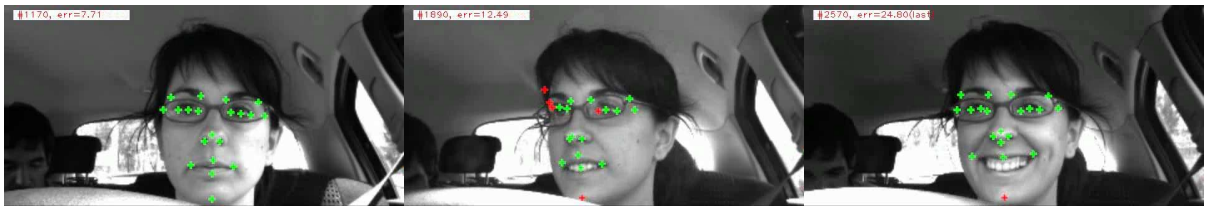
Figure 5.25: Error plots for STASM and R-SMAT in sequence # 7

patch size, although models with patches of 15×15 and 20×20 pixels gave better results.

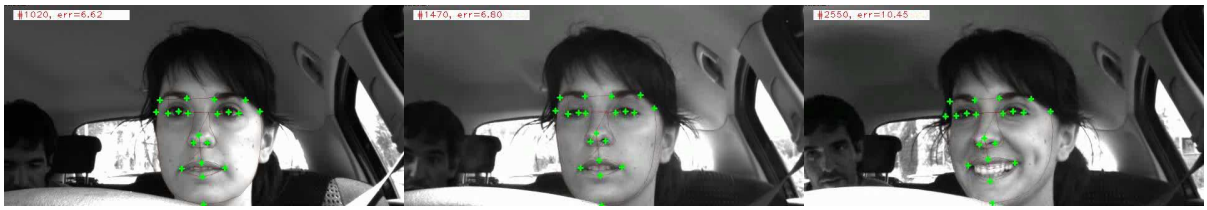
The main contribution of this chapter is enhancing SMAT with alternative clustering methods and a robust shape fitting algorithm. The new method, Robust SMAT (R-SMAT) has shown much better accuracy than the original proposal, and it is robust to occlusions and self-occlusions. Initialized with STASM, it has shown similar performance to this method, with much faster execution timings. Correct tracking has been demonstrated on all types of sequences, for all drivers. Finally, it is able to run in real-time, with average processing rates over 100 fps, including the tracking loss detection and recuperation processes.

A comparison has been made between R-SMAT and STASM in some typical situations that appear in the intended application: head turns, occlusions, driver gesturing and talking, quick illumination changes and nighttime driving, simulated with a low-light environment in the simulator sequences of RS-DMV. R-SMAT performs similarly to STASM when the face is frontal to the camera, and additionally it is robust to head turns.

(a) Seq. #1, Frame 1170 STASM (b) Seq. #1, Frame 1890 STASM (c) Seq. #1, Frame 2570 STASM



(d) Seq. #1, Frame 1170 R-SMAT (e) Seq. #1, Frame 1890 R-SMAT (f) Seq. #1, Frame 2570 R-SMAT



(g) Seq. #2, Frame 1020 STASM (h) Seq. #2, Frame 1470 STASM (i) Seq. #2, Frame 2550 STASM

(j) Seq. #2, Frame 1020 R-SMAT (k) Seq. #2, Frame 1470 R-SMAT (l) Seq. #2, Frame 2550 R-SMAT

Figure 5.26: Drivers wearing glasses: STASM and R-SMAT fitted in sequence #1 and #2

(a) Frame 1200 STASM (b) Frame 1233 STASM (c) Frame 1236 STASM

(d) Frame 1200 R-SMAT (e) Frame 1233 R-SMAT (f) Frame 1236 R-SMAT

(g) Frame 1306 STASM (h) Frame 1312 STASM (i) Frame 1316 STASM

(j) Frame 1306 R-SMAT (k) Frame 1312 R-SMAT (l) Frame 1316 R-SMAT

Figure 5.27: Illumination change: STASM and R-SMAT fitted to sequence #6

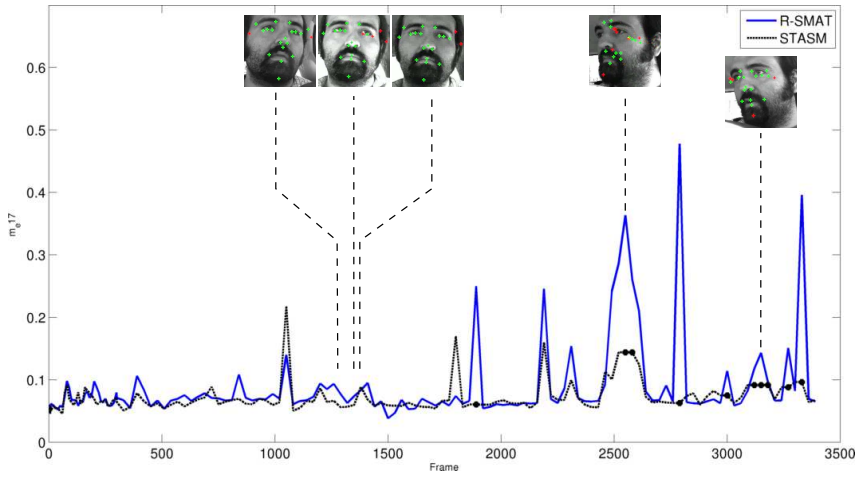
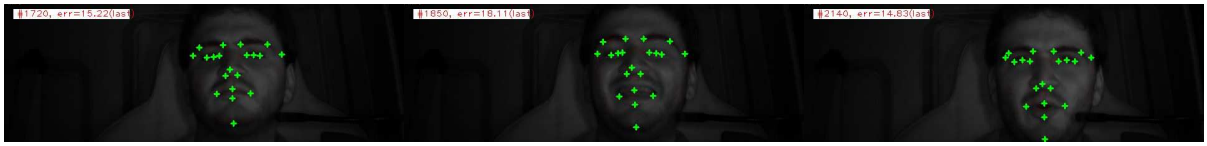
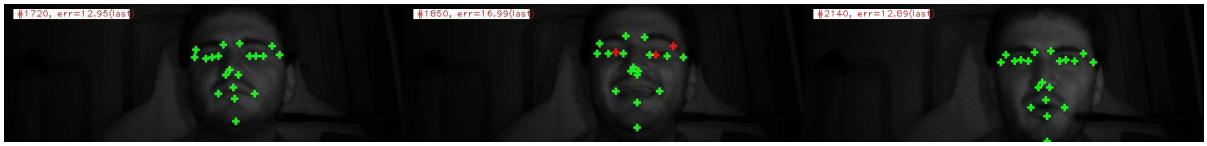


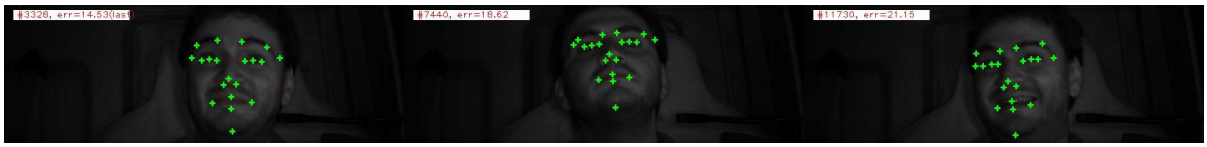
Figure 5.28: Error plots for STASM and R-SMAT in sequence #6



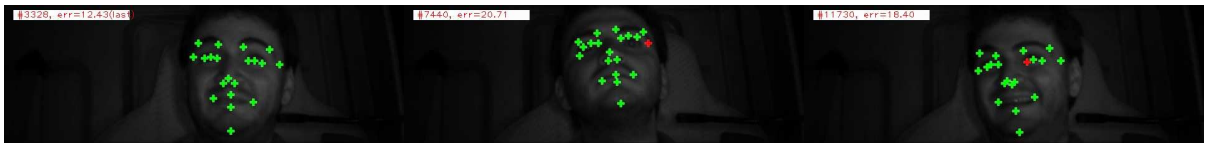
(a) Frame 1720 STASM (b) Frame 1850 STASM (c) Frame 2140 STASM



(d) Frame 1720 R-SMAT (e) Frame 1850 R-SMAT (f) Frame 2140 R-SMAT



(g) Frame 3328 STASM (h) Frame 7440 STASM (i) Frame 11730 STASM



(j) Frame 3328 R-SMAT (k) Frame 7440 R-SMAT (l) Frame 11730 R-SMAT

Figure 5.29: Low light environment: STASM and R-SMAT fitted to sequence #9

Chapter 6

Conclusions and future work

6.1 Conclusions

The starting point of this thesis is the influence of driver inattention in traffic accidents, and how an automatic system could help identify dangerous situations and reduce the number of crashes, saving lives and money. Driver monitoring is a complex task, and involves many parameters of behaviour and physiology. Analyzing head movements and facial expressions and actions like blinking or gaze fixation using computer vision can produce a precise estimation of the state of the driver.

The first levels of such a computer vision system are face localization and tracking. This thesis has focused on the latter, using active models to characterize the face. Three types of active models have been tested on a dataset of videos of drivers in real scenarios and simulators. The focus of these tests have been to show if the models could be used in different situations, and if they comply with the requirements a production system would demand, that is, real-time execution, robustness to head turns and occlusions, facial expressions and varying illumination, and users wearing glasses, among others.

There is, to the best of our knowledge, no available dataset of videos of people driving. A new dataset, RS-DMV, has been compiled, comprised of sequences recorded in a real scenario, in a truck simulator in low light conditions with fully awake drivers presented with dangerous situations that would highlight distractions, and in a passenger car simulator with drivers in fatigue.

The Stacked Trimmed ASM (STASM) [Milborrow 08] integrates a number of extensions to the original Active Shape Model [Cootes 95]. STASM demonstrated great accuracy in all tests when the face was frontal, but did not work when the head was rotated. It has been found to be robust to occlusions, changes in illumination and it is able to work properly in very dark images, typical of night driving. Its other main drawback is that it runs at less than 3 frames per second on our test machine, which makes it impractical for our intended application. Constrained Local Models (CLM) [Cristinacce 06] are able to run in real-time at over than 30 frames per second, but its accuracy is much worst than STASM's, and needs periodic reinitialization to avoid drifting. Again, it is not robust to head turns, and thus it does not comply with the system requirements.

Simultaneous Modeling and Tracking (SMAT) [Dowson 05] characterizes shape and texture as the sequence advances, using samples from previous frames to build the model incrementally. The texture in a patch around each point of the shape is described with a set of clusters. The shape is modeled with its own set of clusters. New clustering algorithms have been proposed to be used in SMAT and have improved both the reliability and the representation capacity of the model, while reducing computational resources. A

shape model built from a training set has been also integrated. Coupled with a robust fitting function (an M-estimator), the robustness of the model to occlusions and head turns improved considerably. We call this new method Robust SMAT (R-SMAT). In the implementation developed for this thesis, R-SMAT is able to run at speeds over 100 frames per second. R-SMAT has been tested on a series of typical situations in the application: head turns, occlusions, illumination changes and facial expressions.

6.2 Main contributions

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

1. **RS-DMV Data set.** Video sequences of drivers in three different scenarios were recorded. A first group of videos was captured in a car driven around the University Campus, with subjects fully alert. A second set was recorded in a truck simulator, in dark conditions that approached those of night driving.Drivers were exposed to dangerous situations that highlighted the effect of distraction. Finally, a third set of videos was recorded in a car simulator, with subjects that exhibited drowsiness. The RS-DMV dataset contains most actions present in everyday driving, and makes an ideal test set for evaluating face tracking algorithms in a driving scenario. The first two sets of videos are to be made available to the research community. The other set is subject to intellectual property restrictions and can not be distributed. We are working towards obtaining the right of distribution for these sequences.

2. **Evaluation of Active Models.** Active Shape Models (ASM) and Constrained Local Models (CLM) have been tested on the RS-DMV dataset. Stacked Trimmed ASM (STASM) was found to have excellent accuracy when the face is frontal, but was not able to work on rotated faces. Performance of CLM was poorer in our sequences than that reported in [Cristinacce 08], and needed frequent re-initialization. On the downside, STASM does not run in real-time, well below the required 30 fps. CLM runs in real-time.

3. **Automatic texture modeling with incremental clustering.** Three additional incremental clustering algorithms have been integrated in SMAT. The first is the classical *leader* algorithm, a modification of it (*leaderP*) and a simple hierarchical method. All three have shown improved the reliability and representation capacity of the model, while reducing computational resources.

4. **Robust shape fitting integrated in SMAT.** The incremental shape model of the original SMAT proposal showed poor results, and was replaced by a model built *a priori*. Huber function (an M-estimator) was used to fit the shape, and the robustness of SMAT to occlusions and head turns improved considerably

5. **Robust SMAT (R-SMAT).** Combining the 3 proposed incremental clustering methods and the robust shape fitting, a more robust and accurate SMAT, called R-SMAT has been developed. Initialized with STASM, it shows similar performance to STASM with much shorter processing times (over 100fps), and works reliably on situations of interest for the intended application of this thesis.

6.3 Future work

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

- **Automatic shape learning.** The current implementation of R-SMAT only models the texture on-line, and still requires a shape model trained off-line. Researching methods of building a robust shape model on-line would eliminate this step and make R-SMAT fully autonomous. The first step is to improve the cluster building procedure so it remove scale and rotation from the shape before making a shape part of a cluster.

- **Model tainting in R-SMAT.** Model tainting is a problem for R-SMAT that has not been solved yet. Better and earlier detection of losses will help, but methods for blocking the insertion of outliers in the model, and the removal of those if they are included, will increase the robustness of R-SMAT after tracking losses.

- **Extending the applications of R-SMAT.** Testing R-SMAT on faces in other scenarios than driving, or on other kinds of deformable objects would provide additional evaluations of R-SMAT. Challenging tests could demonstrate different performance of the clustering algorithms.

- **Multi-scale R-SMAT.** A version of R-SMAT with a multi-scale version of the texture modeling could be of help in scenarios where the face being tracked displaces from or to the camera.

- **Extension of the Dataset.** The sequences on the RS-DMV dataset present most of the actions that take place in normal driving, but more comprehensive and diverse datasets are always desirable.

- **Integration on a driver monitoring system.** The next step to build the intended driver monitoring system is to integrate R-SMAT with additional systems that analyze eye actions and extract parameters (PERCLOS, blinking frequency,...) that can reflect drowsiness, and gaze estimation algorithms. The final step is to install the complete system in a production vehicle.

Appendix A

Software

Several packages of software have been developed in this thesis. Constrained Local Models (CLM), Simultaneous Modeling and Tracking (SMAT) and R-SMAT were all coded from scratch. Additionally, a few helper tools were developed: `Feature Point Marker` (`FPM`) eases the handmarking of images, `cvLoadFS` provides a way of reading OpenCV XML files into MATLAB, `RawVideoTools` manipulates uncompressed video sequences.

All software has been written in C++, and uses the Standard Template Library (STL). OpenCV 1.0 is the main library the software relies on, and it is used for most image processing and mathematical operations. GNU Scientific Library (GSL) implementation of the Nelder-Mead simplex minimization method is used in the CLM code. Boost's smart pointers[1] are widely employed in the code to avoid memory leaks.

OpenCV's functionality for accessing and writing videos (through the FFMPEG library[2]) are used to read unprocessed data (except when video files are in RAW format), and store the results of processing. Displaying of the results is also done with OpenCV's GUI system (based on GTK+ in GNU/Linux). OpenCV's capabilities are very limited in this sense, but they are also easy to use.

This appendix briefly describes the tool developed to allow for faster and better marking of the *keyframes* of the sequences in the RS-DMV dataset. The code of this tool, and others needed to read RAW video files will be released along with the RS-DMV dataset.

A.1 Feature Point Marker

`Feature Point Marker` (`FPM`) was written to make image marking as fast and easy as possible. The functionality that have been added to it derive from past experiences using similar tools the candidate developed and observing their weaknesses.

Reference shape The first requirement for a tool like `FPM` is to have *reference shape*. This shape is a sample of the distribution of the points. When marking images with many points (more than 15), it is of great help to have a guide of where the next point has to be placed: areas that contain many landmarks may confuse the human marker, and points get swapped easily. `FPM` solves this by always drawing a shape in the image, which the human operator can move, scale and deform as required, as in figure A.1. If points are placed too close to each other and discerning the correct shape is not easy, the position of the landmarks can be reset to the default.

[1] `www.boost.org`
[2] `http://ffmpeg.org/`

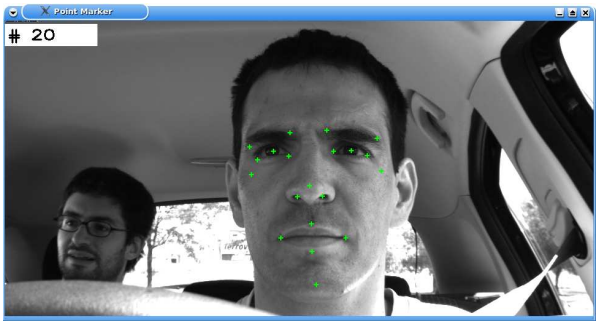
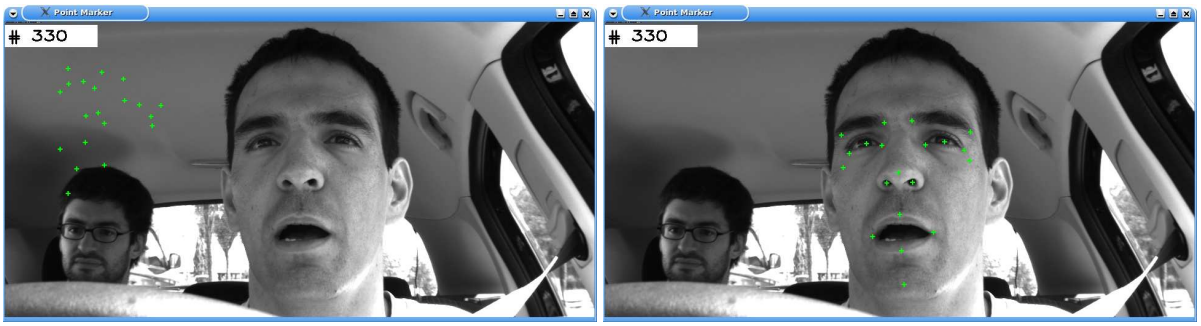


Figure A.1: The default shape, drawn over a frame

Occluded landmarks Sequences in the RS-DMV dataset contain frames where the face is occluded by the subject's hand, or self-occluded in a head turn. Occluded points can be marked as such by pressing a key. The landmark color changes from the default green cross to a white circle inside a black circumference. In the XML file, the position of occluded landmarks in the image is multiplied by -1, so they can be identified as not valid by other software.

Translate, Rotate, Scale Placing the landmarks one by one takes considerable time, and there is a risk of mixing them in the process. `FPM` has shortcuts to translate, rotate and scale the whole shape easily. The points can then be positioned easily over the face. In many cases, many of them can be placed on their correct position, and only a few will have to be moved specifically.



(a) (b)

Figure A.2: Translating and rotating the shape, with a few keystrokes

Groups of landmarks Landmarks that are part of a area like the eyes or the nose will move and deform in similar ways. `FPM` allows the human marker to select groups of landmarks, which can be moved around as if they were just one point, as shown in figure A.3. Adding points to a group is done by pressing the `Ctrl` key and clicking on the landmarks. `Ctrl`+click again will deselect the landmark.

These groups can be linked to a *hotkey*: pressing the *hotkey* will select the whole group. `FPM` supports up to 10 groups of landmarks.

Reuse of past markings Sequences contain many similar frames, and reuse of past markings can reduce the time to handmark a new frame. All past markings of a sequence

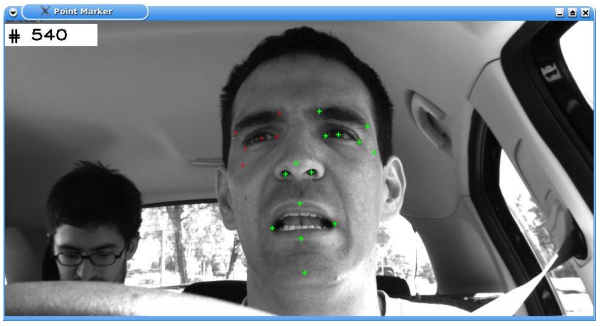


Figure A.3: A group of landmarks selected

can be invoked, until one that is close enough to the current is found.

Marking frequency Ground-truth data for the RS-DMV dataset is given for the first 10 frames of a sequence, then for 1 in every 10 frames up to the 300th frame and then for every 30 frames until the end of the video. `FPM` can be given a parameter to indicate the frequency of frames that have to be marked. It is specified in file containing a CvMat with 3 columns, indicating the first and last frame of a range of images, and the frequency of frames to be marked for the given range. As many ranges as needed can be specified.

Saving and loading of marking sessions FPM saves and loads all data needed to retake a marking session where it was left. These data include the position of the landmarks in each frame, the reference shape, the marking frequency and euclidean transformations of each shape if these are present. Because other software may not need this information, the position of the landmarks alone is saved to another file.

To prevent loss of data in the event of a crash or shutdown, session data is saved automatically after each frame to a temporal file.

Bibliography

[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.

[Asharaf 03] S. Asharaf & M.N. Murty. *An adaptive rough fuzzy single pass algorithm for clustering large data sets*. Pattern Recognition, vol. 36, no. 12, pages 3015–3018, 2003.

[Asharaf 06] S. Asharaf, M. Narasimha Murty & SK Shevade. *Rough set based incremental clustering of interval data*. Pattern Recognition Letters, vol. 27, no. 6, pages 515–519, 2006.

[Avidan 07] S. Avidan. *Ensemble Tracking*. IEEE Trans. Pattern Anal. Mach. Intell., pages 261–271, 2007.

[AWAKE Consortium 04] AWAKE Consortium. *AWAKE-System for Effective Assessment of Driver Vigilance and Warning According to Traffic Risk Estimation*, September 2001-2004. (IST 2000-28062).

[Baker 01] S. Baker & I. Matthews. *Equivalence and efficiency of image alignment algorithms*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, 2001.

[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 & Jeff Schneider. *Automatic Construction of Active Appearance Models as an Image Coding Problem*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 10, pages 1380–1384, October 2004.

[Baker 04c] 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.

[Basu 96] S. Basu, I. Essa & A. Pentland. *Motion regularization for model-based head tracking*. In International Conference on Pattern Recognition, volume 13, pages 611–616, 1996.

[Belhumeur 97] P.N. Belhumeur, J.P. Hespanha & D.J. Kriegman. *Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 7, pages 711–720, July 1997. Special Issue on Face Recognition.

[Bergasa 00] L. M. Bergasa, M. Mazo, A. Gardel, M. A. Sotelo & L. Boquete. *Unsupervised and adaptive Gaussian skin-color model.* Image and Vision Computing, vol. 18, pages 987–1003, sep 2000.

[Bergasa 04] Luis M. Bergasa, Jesús Nuevo, M.A. Sotelo & M. Vazquez. *Real-Time System for Monitoring Driver Vigilance.* In Proc. IEEE Intelligent Vehicles Symposium, pages 78–83, Parma, Italy, June 2004.

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

[Bergasa 08] L.M. Bergasa, J.M. Buenaposada, J. Nuevo, P. Jimenez & L. Baumela. *Analysing Driver's Attention Level using Computer Vision.* In Intelligent Transportation Systems. ITSC 2008. 11th International IEEE Conference on, pages 1149–1154, 2008.

[Birchfield 98] S. Birchfield. *Elliptical head tracking using intensity gradients and color histograms.* In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 232–237, 1998.

[Black 97] M.J. Black & Y. Yacoob. *Recognizing facial expressions in image sequences using local parameterized models of image motion.* International Journal of Computer Vision, vol. 25, no. 1, pages 23–48, 1997.

[Blanz 99a] Volker Blanz & Thomas Vetter. *A morphable model for the synthesis of 3D faces.* In SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[Blanz 99b] Volker Blanz & Thomas Vetter. *A morphable model for the synthesis of 3D faces.* In SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[Bookstein 91] F.L. Bookstein. Morphometric Tools for Landmark Data: Geometry and Biology. Cambridge University Press, 1991.

[Boyraz 08] P. Boyraz, M. Acar & D. Kerr. *Multi-sensor driver drowsiness monitoring.* Proceedings of the Institution of Mechanical En-

gineers, Part D: Journal of Automobile Engineering, vol. 222, no. 11, pages 2041–2062, 2008.

[Bradski 98] G.R. Bradski*et al.* *Computer vision face tracking for use in a perceptual user interface.* Intel Technology Journal, vol. 2, no. 2, pages 12–21, 1998.

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

[Buchsbaum 80] G. Buchsbaum. *A spatial processor model for object colour perception.* J. Franklin inst, vol. 310, no. 1, pages 1–26, 1980.

[Buenaposada 01] J. M. Buenaposada, D. Sopena & L. Baumela. *Face tracking using the dynamic grey world algorithm.* Lecture notes in computer science, pages 341–348, 2001.

[Buenaposada 04] J. M. Buenaposada, E. Muñoz & L. Baumela. *Efficient appearance-based tracking.* In Computer Vision and Pattern Recognition Workshop, 2004 Conference on, pages 6–6, 2004.

[Buenaposada 06] J. M. Buenaposada, E. Muñoz & L Baumela. *Efficiently estimating facial expression and illumination in appearance-based tracking.* In Proc. British Machine Vision Conference, volume 1, pages 57–66, 2006.

[Buenaposada 08] J. M. Buenaposada, E. Muñoz & L. Baumela. *Recognising facial expressions in video sequences.* Pattern Analysis and Applications, vol. 11, no. 1, pages 101–116, 2008.

[Burel 94] G. Burel & D. Carel. *Detection and localization of faces on digital images.* Pattern Recognition Letters, vol. 15, no. 10, pages 963–967, 1994.

[Burl 95] MC Burl, TK Leung & P. Perona. *Face localization via shape statistics.* In Proc. First Inter. Workshop on Automatic Face and Gesture Recognition, pages 154–159, 1995.

[Can 93] F. Can. *Incremental clustering for dynamic information processing.* ACM Transactions on Information Systems (TOIS), vol. 11, no. 2, pages 143–164, 1993.

[Chai 99] D. Chai & K.N. Ngan. *Face segmentation using skin-color map in videophone applications.* Circuits and Systems for Video Technology, IEEE Transactions on, vol. 9, no. 4, pages 551–564, Jun 1999.

[Charikar 97] M. Charikar, C. Chekuri, T. Feder & R. Motwani. *Incremental clustering and dynamic information retrieval.* In Proceedings of the 20th annual ACM symposium on Theory of computing, pages 626–635. ACM New York, NY, USA, 1997.

[Chetverikov 92] Dmitry Chetverikov & Attila Lerch. *Multiresolution feca detection.* In Theoretical Foundations of Computer Vision, 1992.

[Clark 04] Adrian F. Clark & Christine Clark. *Performance Characterization in Computer Vision: A Tutorial*. Tech. rep., VASE Laboratory, Electronic Systems Engineering, University of Essex, 2004.

[Collins 05] R.T. Collins, Y. Liu & M. Leordeanu. *Online Selection of Discriminative Tracking Features*. IEEE Trans. Pattern Anal. Mach. Intell., pages 1631–1643, 2005.

[Cootes 92] T.F. Cootes & C.J. Taylor. *Active shape models–smart snakes*. In Proc. British Machine Vision Conference, 1992.

[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. Mach. Intell., vol. 23, pages 681–685, January 2001.

[Cootes 01b] TF Cootes & CJ Taylor. *On representing edge structure for model matching*. Proc. IEEE CVPR, vol. 1, pages 1114–1119, 2001.

[Cootes 02] TF Cootes, GV Wheeler, KN Walker & CJ 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.

[Craw 87] J. Craw, H. Ellis & JR Lishman. *Automatic extraction of face-features*. Pattern Recognition Letters, vol. 5, no. 2, pages 183–187, 1987.

[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.

[Cristinacce 07] D. Cristinacce & T. Cootes. *Boosted regression active shape models*. In 18 th British Machine Vision Conference, Warwick, UK, pages 880–889, 2007.

[Cristinacce 08] D. Cristinacce & T. Cootes. *Automatic feature localisation with constrained local models*. Pattern Recognition, 2008.

[Crowley 97] James L. Crowley & Francois Berard. *Multi-Modal Tracking of Faces for Video Communications*. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 0, page 640, 1997.

[DaimerAG 01] DaimerAG. *The electronic drawbar*, June 2001. http://www.daimler.com.

[DaimerAG 09] DaimerAG. *Attention Assist*, June 2009. http://www.daimler.com.

[Delaunay 34] B. Delaunay. *Sur la sphere vide*. Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk, vol. 7, pages 793–800, 1934.

[Dinges 98] D. Dinges. *F. PERCLOS: A valid psychophysiological measure of alertness as assesed by psychomotor vigilance*. Tech. Rep. MCRT-98-006, Federal Highway Administration. Office of motor carriers, 1998.

[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*. Tech. rep., Virginia Tech Transportation Institute, NHTSA, April 2006.

[D'Orazio 07] T. D'Orazio, M. Leo, C. Guaragnella & A. Distante. *A visual approach for driver inattention detection*. Pattern Recognition, vol. 40, no. 8, pages 2341–2355, 2007.

[Dornaika 04] F. Dornaika & J. Ahlberg. *Fast and reliable active appearance model search for 3-d face tracking*. IEEE Trans. Syst., Man, Cybern. B, vol. 34, no. 4, pages 1838–1853, 2004.

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

[Dowson 06] N. Dowson & R. Bowden. *N-tier simultaneous modelling and tracking for arbitrary warps*. In Proc. of the 17th British Machine Vision Conference. British Machine Vision Association, volume 1, page 6, 2006.

[Dryden 98] I.L. Dryden & K.V. Mardia. Statistical shape analysis. John Wiley & Sons, 1998.

[Ekman 78] P. Ekman, W.V. Friesen, J.C. Hager & A.H. Face. Facial Action Coding System. Consulting Psychologists Press, 1978.

[European Commision 03] European Commision. *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.

[Felzenszwalb 05] P.F. Felzenszwalb & D.P. Huttenlocher. *Pictorial structures for object recognition*. International Journal of Computer Vision, vol. 61, no. 1, pages 55–79, 2005.

[FGNet 04] FGNet. *Face and Gesture Recognition Working group*, 2004. http://www-prima.inrialpes.fr/FGnet/html/home.html.

[Fisher 87] D.H. Fisher. *Knowledge acquisition via incremental conceptual clustering*. Machine Learning, vol. 2, no. 2, pages 139–172, 1987.

[Flatley 04] D. Flatley, L.A. Reyner & J.A. Horne. *Sleep-related vehicle crashes - the relationship to traffic density*. In Behavioural research in road safety: 13th seminar proceedings. Sleep Research Centre, Loughborough University, January 2004.

[FRAV 09] Face Recognition & Artificial Vision Group FRAV. *FRAV3D Database*. Universidad Rey Juan Carlos, Madrid, 2009. http://www.frav.es/databases/FRAV3D/.

[FSF 91] Free Software Foundation FSF. *GNU General Public License*. online, http://www.gnu.org/copyleft/gpl.html, 1991.

[Gee 94] A.H. Gee & R. Cipolla. *Non-intrusive gaze tracking for human-computer interaction*. In Proc. Mechatronics and Machine Vision in Practise, pages 112–117. Citeseer, 1994.

[Gorodnichy 02] D.O. Gorodnichy. *On importance of nose for face tracking*. In Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG 2002), pages 20–21, 2002.

[Gower 75] J.C. Gower. *Generalized procrustes analysis*. Psychometrika, vol. 40, no. 1, pages 33–51, 1975.

[Grabner 06] H. Grabner & H. Bischof. *On-line boosting and vision*. In Proc. CVPR, volume 1, pages 260–267, 2006.

[Gross 01] Ralph Gross, Jianbo Shi & Jeffrey Cohn. *Quo Vadis Face Recognition?* In Third Workshop on Empirical Evaluation Methods in Computer Vision, December 2001.

[Gross 04] Ralph Gross, Iain Matthews & Simon Baker. *Constructing and Fitting Active Appearance Models With Occlusion*. In Proceedings of the IEEE Workshop on Face Processing in Video, June 2004.

[Gross 05a] Ralph Gross. *Face Databases*. In A.Jain S.Li, ed., Handbook of Face Recognition. Springer, New York, February 2005.

[Gross 05b] Ralph Gross, Iain Matthews & Simon Baker. *Generic vs. person specific active appearance models*. Image and Vision Computing, vol. 23, no. 11, pages 1080–1093, November 2005.

[Gunn 94] SR Gunn & MS Nixon. *A dual active contour for head boundary extraction.* In IEE Colloquium on Image Processing for Biometric Measurement, page 6, 1994.

[Hager 98] Gregory D. Hager & Peter N. Belhumeur. *Efficient Region Tracking With Parametric Models of Geometry and Illumination.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 10, pages 1025–1039, 1998.

[Hansen 05] D.W. Hansen & A.E.C. Pece. *Eye tracking in the wild.* Computer Vision and Image Understanding, vol. 98, no. 1, pages 155–181, 2005.

[Hartigan 75] J.A. Hartigan. Clustering Algorithms. John Wiley & Sons, Inc. New York, NY, USA, 1975.

[Hjelmas 01] E. Hjelmas & B.K. Low. *Face detection: A survey.* Computer Vision and Image Understanding, vol. 83, no. 3, pages 236–274, 2001.

[Huang 07] C. Huang, H. Ai, Y. Li & S. Lao. *High-performance rotation invariant multiview face detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 4, pages 671–686, 2007.

[Huber 81] Peter J. Huber. Robust statistics. Wiley Series in Probability and Mathematical Statistics. Wiley-Interscience, 1981.

[Jain 99] AK Jain, MN Murty & PJ Flynn. *Data Clustering: A Review.* ACM Computing Surveys, vol. 31, no. 3, 1999.

[Jebara 97] T. Jebara & A. Pentland. *Parametrized structure from motion for 3d adaptive feedback tracking of faces.* In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 144–150. IEEE, 1997.

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

[Jesorsky 01] O. Jesorsky, K.J. Kirchberg, R. Frischholz*et al. Robust face detection using the hausdorff distance.* Proceedings of Audio and Video based Person Authentication, pages 90–95, 2001. `http://www.bioid.com/downloads/facedb/`.

[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, Oct 2002.

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

[Jolliffe 02] I.T. Jolliffe. Principal Component Analysis. Springer, 2002.

[Jones 02] M.J. Jones & J.M. Rehg. *Statistical color models with application to skin detection*. International Journal of Computer Vision, vol. 46, no. 1, pages 81–96, 2002.

[Jurie 02] F. Jurie & M. Dhome. *Hyperplane approximation for template matching*. IEEE Trans. Pattern Anal. Mach. Intell., pages 996–1000, 2002.

[Kanade 98] Takeo Kanade, Hideo Saito & Sundar Vedula. *The 3D Room: Digitizing Time-Varying 3D Events by Synchronized Multiple Video Streams*. Tech. Rep. CMU-RI-TR-98-34, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1998.

[Kanade 00] T. Kanade, J.F. Cohn & Y. Tian. *Comprehensive database for facial expression analysis*. Proceedings of the fourth IEEE International conference on automatic face and gesture recognition (FG'00), pages 46–53, 2000.

[Kaneko 02] T. Kaneko & O. Hori. *Template update criterion for template matching of image sequences*. In International Conference on Pattern Recognition, volume 2, 2002.

[Kass 88] M. Kass, A. Witkin & D. Terzopoulos. *Snakes: Active contour models*. International journal of computer vision, vol. 1, no. 4, pages 321–331, 1988.

[Kendall 84] D.G. Kendall. *Shape manifolds, procrustean metrics, and complex projective spaces*. Bulletin of the London Mathematical Society, vol. 16, no. 2, page 81, 1984.

[Kim 98] S.H. Kim, N.K. Kim, S.C. Ahn & H.G. Kim. *Object oriented face detection using range and color information*. In Proc. Third Int'l Conf. Automatic Face and Gesture Recognition, pages 76–81, 1998.

[Kircher 02] Albert Kircher, Marcus Uddman & Jesper Sandin. *Vehicle control and drowsiness*. Tech. Rep. VTI-922A, Swedish National Road and Transport Research Institute, 2002.

[Klein 07] S.B. Klein & B.M. Thorne. Biological Psychology. Worth Pub, 2007.

[Kleiner 04] Mario Kleiner, Christian Wallraven & Heinrich H. Bülthoff. *The MPI VideoLab - A system for high quality synchronous recording of video and audio from multiple viewpoints*. Tech. rep., Max Planck Institute for Biological Cybernetics, may 2004.

[Kutila 06] Matti Kutila. *Methods for Machine Vision Based Driver Monitoring Applications*. PhD thesis, VTT Technical Research Centre of Finland, 2006.

[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 3-D models*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 4, pages 322–336, 2000.

[Lal 02] S.K.L. Lal & A. Craig. *Driver fatigue: Electroencephalography and psychological assessment*. Psychophysiology, vol. 39, no. 03, pages 313–321, 2002.

[Lam 94] K.M. Lam & H. Yan. *Fast algorithm for locating head boundaries (Journal Paper)*. Journal of Electronic Imaging, vol. 3, no. 04, pages 351–359, 1994.

[Lanitis 95] A. Lanitis, CJ Taylor & TF Cootes. *Automatic face identification system using flexible appearance models*. Image and Vision Computing, vol. 13, no. 5, pages 393–401, 1995.

[Le Gallou 06] S. Le Gallou, G. Breton, C. Garcia & R. Seguier. *Distance maps: A robust illumination preprocessing for active appearance models*. In VISAPP'06, International Conference on Computer Vision Theory and Applications, 2006.

[Li 93] H. Li, P. Roivainen & R. Forchheimer. *3-D motion estimation in model-based facial image coding*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no. 6, pages 545–555, 1993.

[Li 04] Y. Li. *On incremental and robust subspace learning*. Pattern Recognition, vol. 37, no. 7, pages 1509–1518, 2004.

[Liu 03] Y. Liu, K.L. Schmidt, J.F. Cohn & S. Mitra. *Facial asymmetry quantification for expression invariant human identification*. Computer Vision and Image Understanding, vol. 91, no. 1-2, pages 138–159, 2003.

[Lucas 81] B.D. Lucas & T. Kanade. *An iterative image registration technique with an application to stereo vision*. In International Joint Conference on Artificial Intelligence, volume 3, pages 674–679, 1981.

[Mahieu 09] Yves Mahieu. Highlights of the panorama of transport. eurostats, 2009.

[Manning 08] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze. Introduction to information retrieval. Cambridge University Press, 2008.

[Martinez 99] A. M. Martinez & R. Benavente. *The AR face database*. Tech. Rep. 24, Computer Vision Center (CVC), UAB, 1999. http://cobweb.ecn.purdue.edu/ aleix/aleix_face_DB.html.

[Martinkauppi 02] B. Martinkauppi. *Face colour under varying illumination: analysis and applications*. PhD thesis, Department of Electrical and Information Engineering, University of Oulu, 2002.

[Matsumoto 00] Y. Matsumoto & A. Zelinsky. *An algorithm for real-time stereo vision implementation of head pose and gaze direction measurements*. In Procs. IEEE 4th Int. Conf. Face and Gesture Recognition, pages 499–505, mar 2000.

[Matthews 03] Iain Matthews, Takahiro Ishikawa & Simon Baker. *The Template Update Problem*. In Proceedings of the British Machine Vision Conference, September 2003.

[Matthews 04a] I. Matthews, T. Ishikawa & S. Baker. *The Template Update Problem*. IEEE Trans. Pattern Anal. Mach. Intell., pages 810–815, 2004.

[Matthews 04b] Iain Matthews & Simon Baker. *Active Appearance Models Revisited*. International Journal of Computer Vision, vol. 60, no. 2, pages 135–164, November 2004.

[McKenna 95] S. McKenna, S. Gong & H. Liddell. *Real-time tracking for an integrated face recognition system*. In Second European Workshop on Parallel Modelling of Neural Operators, volume 11. Citeseer, 1995.

[McKenna 98] S.J. McKenna, S. Gong & Y. Raja. *Modelling facial colour and identity with gaussian mixtures*. Pattern Recognition, vol. 31, no. 12, pages 1883–1892, 1998.

[McKenna 99] S.J. McKenna, Y. Raja & S. Gong. *Tracking colour objects using adaptive mixture models*. Image and Vision Computing, vol. 17, no. 3-4, pages 225–231, 1999.

[McLachlan 97] G.J. McLachlan & T. Krishnan. The EM algorithm and extensions. Wiley New York, 1997.

[Messer 99] K. Messer, J. Matas, J. Kittler, J. Luettin & G. Maitre. *XM2VTSDB: The extended M2VTS database*. In Second International Conference on Audio and Video-based Biometric Person Authentication, volume 964, pages 965–966. Citeseer, 1999.

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

[MISKL 05] MISKL. *The BJUT-3D Large-Scale Chinese Face Database*. Tech. rep., Multimedia and Intelligent Software Technology Beijing Municipal Key Laboratory, Beijing University of Technology, Aug 2005.

[Missouri DoT 07] Missouri DoT. *Traffic Crash Statistics*. Tech. rep., Missouri Department of Transportation, 2007.

[Morency 03] L.P. Morency, A. Rahimi & T. Darrell. *Adaptive view-based appearance models*. In 2003 IEEE Computer Society Conference

on Computer Vision and Pattern Recognition, 2003. Proceedings, volume 1, 2003.

[Nelder 65] J.A. Nelder & R. Mead. *A simplex method for function minimization.* Computer Journal, vol. 7, no. 4, pages 308–313, 1965.

[Osuna 97] E. Osuna, R. Freund & F. Girosit. *Training support vector machines: an application to face detection.* In 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings., pages 130–136, 1997.

[O'Toole 05] A.J. O'Toole, J. Harms, S.L. Snow, D.R. Hurst, M.R. Pappas, J.H. Ayyad & H. Abdi. *A video database of moving faces and people.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 5, pages 812–816, May 2005.

[Paterson 03] J. Paterson & A. Fitzgibbon. *3d head tracking using non-linear optimization.* In Proc. BMVC, pages 609–618, 2003.

[Perez 02] P. Perez, C. Hue, J. Vermaak & M. Gangnet. *Color-based probabilistic tracking.* Lecture Notes in Computer Science, pages 661–675, 2002.

[Pham 07a] M.T. Pham & T.J. Cham. *Fast training and selection of haar features using statistics in boosting-based face detection.* In Proceedings of the International Conference on Computer Vision, pages 1–7, 2007.

[Pham 07b] M.T. Pham & T.J. Cham. *Online Learning Asymmetric Boosted Classifiers for Object Detection.* In Computer Vision and Pattern Recognition CVPR, pages 1–8, 2007.

[Pilet 05] J. Pilet, V. Lepetit & P. Fua. *Real-Time Non-Rigid Surface Detection.* In IEEE Conference on Computer Vision and Pattern Recognition 2007, San Diego, CA, June 2005.

[Rasmussen 92] Edie Rasmussen. Clustering algorithms, pages 419–442. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[Rau 05] Paul Rau. *Drowsy driver detection and warning system for commercial vehicle drivers: Field operational test design, analysis and progress.* Tech. rep., NHTSA, 2005.

[Rechtschaffen 98] A. Rechtschaffen. *Current perspectives on the function of sleep.* Perspectives in biology and medicine, vol. 41, no. 3, pages 359–390, 1998.

[Rogers 02] M. Rogers & J. Graham. *Robust active shape model search.* Lecture Notes in Computer Science, pages 517–530, 2002.

[Romdhani 01] S. Romdhani, P. Torr, B. Scholkopf & A. Blake. *Computationally efficient face detection.* In Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings, volume 2, 2001.

[Rowley 96] H.A. Rowley, S. Baluja & T. Kanade. *Neural network-based face detection*. In Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on, pages 203–208, Jun 1996.

[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.

[Royal 03] Dawn Royal. *Volume I - Findings; National Survey on Distracted and Driving Attitudes and Behaviours, 2002*. Tech. Rep. DOT HS 809 566, The Gallup Organization, March 2003.

[SafetyNet 08] SafetyNet. Annual statistical report. European Road Safety Observatory, 2008. `www.erso.eu`.

[Samal 95] S. Samal & PA Iyengar. *Human face detection using silhouettes*. International journal of pattern recognition and artificial intelligence, vol. 9, no. 6, pages 845–867, 1995.

[Schneiderman 04] H. Schneiderman & T. Kanade. *Object detection using the statistics of parts*. International Journal of Computer Vision, vol. 56, no. 3, pages 151–177, 2004.

[Seeing Machines 04] Seeing Machines. *FaceLAB*, August 2004.

[Seeing Machines 07] Seeing Machines. *Driver State Sensor*, August 2007. http://www.seeingmachines.com/dss.html.

[Segvic 06] S. Segvic, A. Remazeilles & F. Chaumette. *Enhancing the point feature tracker by adaptive modelling of the feature support*. In European Conf. on Computer Vision, ECCV'2006, volume 3952 of *Lecture Notes in Computer Science*, pages 112–124, Graz, Austria, May 2006.

[SENSATION 07] SENSATION. *Advanced sensor development for attention, stress, vigilance and sleep/wakefulness monitoring (SENSATION)*, 2004–2007. European Project FP6 (IST-2002-2.3.1.2). http://www.sensation-eu.org.

[Shih 00] W. Shih & Liu. *A calibration-free gaze tracking technique*. In Proc. 15th Conf Patterns Recognition, volume 4, pages 201–204, Barcelona, Spain, 2000.

[Sim 03] Terence Sim, Simon Baker & Maan Bsat. *The CMU Pose, Illumination, and Expression Database*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 12, pages 1615 – 1618, December 2003.

[SmartEyeAG 09] SmartEyeAG. *AntiSleep*, 2009. `www.smarteye.se`.

[Spath 80] H. Spath & V. Bull. Cluster analysis algorithms for data reduction and classification of objects. Ellis Horwood, 1980.

[Stegmann 02] M.B. Stegmann & D.D. Gomez. *A brief introduction to statistical shape analysis*. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, page 15, 2002.

[Stegmann 03a] MB Stegmann, BK Ersboll & R. Larsen. *FAME-a flexible appearance modeling environment*. Medical Imaging, IEEE Transactions on, vol. 22, no. 10, pages 1319–1331, 2003.

[Stegmann 03b] M.B. Stegmann & R. Larsen. *Multi-band modelling of appearance*. Image and Vision Computing, vol. 21, no. 1, pages 61–67, 2003.

[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.

[Strom 99] J. Strom, T. Jebara, S. Basu & A. Pentland. *Real time tracking and modeling of faces: An ekf-based analysis by synthesis approach*. In Modelling People Workshop, ICCV, 1999.

[Susmáková 04] K. Susmáková. *Human Sleep and Sleep EEG*. Measurement Science Review, vol. 4, 2004.

[Tian 00] Y. Tian, T. Kanade & JF Cohn. *Dual-state parametric eye tracking*. In Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000. Proceedings, pages 110–115, 2000.

[Tian 01] Y. Tian, T. Kanade & J.F. Cohn. *Recognizing Action Units for Facial Expression Analysis*. IEEE Trans. Pattern Anal. Mach. Intell., pages 97–115, 2001.

[Tobii 04] Tobii, 2004. Tobii Technologies AB, `www.tobii.com`.

[Tsalakanidou 05] F. Tsalakanidou, S. Malassiotis & M.G. Strintzis. *Face localization and authentication using color and depth images*. Image Processing, IEEE Transactions on, vol. 14, no. 2, pages 152–168, February 2005.

[Tsapatsoulis 00] N. Tsapatsoulis, Y. Avrithis & S. Kollias. *Efficient face detection for multimedia applications*. In Image Processing, 2000. Proceedings. 2000 International Conference on, volume 2, 2000.

[U. of California 97] U. of California & Sleep Research Society. *Basics of sleep behavior*, 1997. Retrieved July 24th, 2009, http://www.sleephomepages.org/sleepsyllabus.

[Ueno 94] H. Ueno, M. Kaneda & M. Tsukino. *Development of drowsiness detection system*. In Proceedings of Vehicle Navigation and Information Systems Conference, pages 15–20, 1994.

[UN-ECE 07] Statistics of road traffic accidents in europe and north america, volume LI. United Nations, Economic Commission For Europe, 2007.

[Uzümcü 03] M. Uzümcü, AF Frangi, M. Sonka, JHC Reiber & BPF Lelieveldt. *ICA vs. PCA active appearance models: Application to cardiac MR segmentation*. In Proceedings of Medical Image Computing and Computer-Assisted Intervention MICCAI, volume 3, pages 451–458. Springer, 2003.

[Victor 01] T. Victor, O. Blomberg & A. Zelinsky. *Automating the measurement of driver visual behaviours using passive stereo vision*. In Procs. Int. Conf. Series Vision in Vehicles VIV9, Brisbane, Australia, aug 2001.

[Vijaya 04] P. A. Vijaya, M. Narasimha Murty & D. K. Subramanian. *Leaders-Subleaders: An efficient hierarchical clustering algorithm for large data sets*. Pattern Recognition Letters, vol. 25, no. 4, pages 505 – 513, 2004.

[Viola 02] P. Viola & M. Jones. *Fast and robust classification using asymmetric adaboost and a detector cascade*. Advances in neural information processing systems, vol. 2, pages 1311–1318, 2002.

[Viola 04] P. Viola & M.J. Jones. *Robust real-time face detection*. International Journal of Computer Vision, vol. 57, no. 2, pages 137–154, 2004.

[Volvo Car Corp. 08] Volvo Car Corp. *Driver Alert Control*, 2008. `http://www.volvocars.com`.

[Wang 03] J.J.L. Wang & S. Singh. *Video analysis of human dynamics-a survey*. Real-time imaging, vol. 9, no. 5, pages 321–346, 2003.

[Wierwille 96] W. Wierwille, L. Tijerina, S. Kiger, T. Rockwell, E. Lauber & A. Bittne. *Final Report Supplement – Task 4: Review of Workload and Related Research*. Tech. Rep. DOT HS 808 467(4), USDOT, oct 1996.

[Wikipedia 08] Wikipedia. *Depth of Field*. online, http://en.wikipedia.org/wiki/Depth_of_field, August 2008. Wikipedia, the free encyclopedia.

[Willett 88] P. Willett. *Recent trends in hierarchic document clustering: a critical review*. Information Processing and Management: an International Journal, vol. 24, no. 5, pages 577–597, 1988.

[Wu 99] H. Wu, Q. Chen & M. Yachida. *Face detection from color images using a fuzzy pattern matching method*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 6, page 557, 1999.

[Yang 96] J. Yang & A. Waibel. *A real-time face tracker*. In Proceedings of WACV, volume 96, pages 142–147, 1996.

[Yang 02] M.H. Yang, DJ Kriegman & N. Ahuja. *Detecting faces in images: a survey*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 1, pages 34–58, 2002.

[Yang 08] Ming-Hsuan Yang. Encyclopedia of biometrics, chap. Face Detection. Springer, 2008.

[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.

[Yow 97] K.C. Yow & R. Cipolla. *Feature-based human face detection.* Image and Vision Computing, vol. 15, no. 9, pages 713–735, 1997.

[Zhang 97] Zhengyou Zhang. *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting.* Image and Vision Computing Journal, 1997.

[Zhou 02] S. Zhou & R. Chellappa. *Probabilistic human recognition from video.* Lecture Notes In Computer Science, pages 681–697, 2002.