Public University of Navarre

Electrical and Electronic Engineering Department

Contributions to Head Pose Estimation Methods

PhD. dissertation prepared to obtain the Doctor degree

Mikel Ariz Galilea

Advisors:

Rafael Cabeza Laguna

Arantxa Villanueva Larre

Pamplona, April 2016

upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Maite dudan jendeari

upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Agradecimientos

Estas líneas van más allá de la tesis doctoral; representan el agradecimiento a todas esas personas que han formado parte, en mayor o menor medida, de una bonita fase de mi vida que ha dado como resultado este trabajo.

Gracias, en primer lugar, a Rafa y Arantxa, por todas vuestras ideas, por vuestro apoyo constante y, sobre todo, por enseñarme a pensar. Gracias, Rafa, porque has supuesto un ejemplo para mí; por hacer que hablar contigo sea fácil, y porque a menudo tus sabios consejos han ido más allá de la tesis y me han ayudado a tomar decisiones en la vida. Gracias, Arantxa, por estar siempre dispuesta a ayudar; porque tu buen humor resulta imprescindible a lo largo de una tesis, y porque tus ideas a menudo son el contrapunto perfecto a las de Rafa.

Gracias al grupo de Upnianos: Luismi, Coro, Mikel, Jes y Vic. Gracias por hacer que todas esas horas pasadas en la universidad hayan sido un periodo feliz para mí; porque hemos creado una amistad que durará mucho más que una tesis, que ya es decir. Gracias, Fermín, porque apareciste cuando todo el mundo se iba y has hecho que disfrutara tanto o más la parte final de la tesis; gracias por esa visión de la vida que tanto comparto, por las risas y por esas apasionantes conversaciones que nunca nos llevaban a ninguna parte, aunque nos daba igual. Y gracias también a Juanjo, Leo y Laura, compañeros de grupo con los que he podido contar cuando lo he necesitado, y compartir descansos y otros buenos ratos.

Gracias a mis nuevos compañeros del CIMA, porque los cambios nunca son fáciles y vosotros habéis hecho que el mío lo sea. Gracias, Carlos, por darme la oportunidad de trabajar en un campo apasionante en el que cada día aprendo cosas nuevas.

Gracias a los amigos de la cuadrilla, Urtzi, Marcos, Andrés, Gaizka y Andoni, por hacerme desconectar con esa facilidad; por buscarle el lado cómico a mi tesis y hacerme reír tantas veces; porque hablar de enchufes transatlánticos con vosotros no tiene precio; porque con vosotros he recorrido un camino increíble que

comenzó mucho antes de la tesis; porque, con amigos así, todo es más sencillo. Mila esker Urtzi, zuri bereziki, lehenengo pertsonan bizi izan dituzulako nire eguneroko arazo eta arrakasta txikiak; zure hitz lasaiek ikaragarrizko balioa dutelako niretzat; beti hor egon zarelako.

Gracias Marisa, porque tu preocupación y tus ánimos son muy importantes; porque eres como una madre para mí. Gracias Joaquintxo, y gracias Txutxo, porque sois familia y disfruto todo el cariño y apoyo que me dais siempre. Y gracias también Tomás, porque para mí ver la pasión y el interés que tienes por todo lo que hago es el ejemplo de lo que me gustaría ser a tu edad; porque, al fin y al cabo, eres el abuelo que habría querido tener.

Gracias, Cori, por todo. Tú eres la principal responsable de que guarde tan buen recuerdo de este periodo, has recorrido este camino conmigo y lo has hecho inmejorable. Una tesis sin tus sonrisas y ánimos tiene que ser realmente dura, así que por eso y por todo lo demás, muchísimas gracias.

Y, finalmente, gracias aitas. Supongo que unos padres siempre están ahí para todo, pero lo vuestro me parece inigualable. Os lo he intentado agradecer cada vez que he tenido la oportunidad y, aun así, creo que nunca sería suficiente. Os lo debo todo, así que gracias de todo corazón.

Abstract

Head Pose Estimation (HPE) is currently a growing research field, mainly because of the proliferation of human-computer interfaces (HCI) in the last decade. It offers a wide variety of applications, including driver assistance systems, pose-invariant face recognition, human behavior analysis, or popular HCI applications such as gaze estimation systems. HCIs show an increasing tendency to integrate HPE as another bridge for interaction, since it is a rich form of communication. For instance, gaze tracking systems suffer in unconstrained environments because they are very sensitive to head motion, and HPE has become a key point for successful gaze estimation. This thesis thus aims to contribute to the development of robust and accurate HPE methods based on 2D tracking of the face in videos.

With the idea of achieving a better understanding of every aspect of the HPE process, a complete framework has been created in the first part of the thesis as a pillar to sustain the rest of the work. This framework consists of both simulation and realistic environments for HPE algorithm analysis. It includes the recording of two head pose databases of videos, one with synthetically generated heads and the other one with real subjects. They have proven to be extremely useful tools for the purpose, and therefore we expect to make them available for the whole scientific community.

The problem of 3D face reconstruction using only 2D images from the videos has received special attention. A whole chapter has been devoted to the study and comparison of different single-view and multi-view based reconstruction methods in a controlled simulation environment. This has allowed us to isolate the 3D model fitting problem, thus drawing several conclusions regarding the influence of this critical part in a HPE system.

With the aim of achieving a wider impact with this thesis, the pose estimation problem is addressed from a general perspective in which techniques that are

generalizable to any kind of 3D object are proposed. Starting from a basic pose estimation approach (2D tracking & POSIT), different alternatives have been developed to improve performance. On the one hand, a tracking accuracy index (TAI) calculation method has been proposed, based on invariant shape metrics obtained from interlandmark relationships. This allows us to apply weights that compensate for 2D tracking inaccuracies and optimize the 3D pose estimation. On the other hand, outlier detection and outlier correction methods that aim to improve the 2D tracking itself have been proposed, addressing the typical *drifting* problem of point-tracking systems, and hence improving the 3D pose estimation further. These global methods have then been specifically adapted to HPE and evaluated using two head pose databases: our real database, which reflects the expected performance in current technological conditions, and the BU database, a widely referenced older database that allows an extensive comparison with other state-of-the-art HPE methods.

Table of Contents

List of Acronyms

Acronym	Significance
3DMM	3D Morphable Model
AAM	Active Appearance Model
AF	All Frames
ALS	Alternated Least Squares
AOM	Active Orientation Model
ASM	Active Shape Model
AVAM	Adaptive View-based Appearance Model
BA	Bundle Adjustment
BFM	Basel Face Model
BME	Biased Manifold Embedding
BU	Boston University
CHM	Cylindrical Head Model
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DOF	Degrees of Freedom
DVF	Distance Vector Fields
EM	Expectation Maximization
GEM	Generic Elastic Model

GSDM	Global Supervised Descent Method
HCI	Human-Computer Interface
HPE	Head Pose Estimation
HPR	Hidden Point Removal
IF	IntraFace
IRLS	Iteratively Re-weighted Least Squares
LDQP	Local Directional Quaternary Pattern
LE	Laplacian Eigenmap
LEA	Locally Embedded Analysis
LLE	Locally Linear Embedding
LM	Levenberg-Marquardt
LPP	Locality Preserving Projections
MLP	Multilayer Perceptron
NLS	Nonlinear Least Squares
OC	Outlier Correction
OD	Outlier Detection
OF	Outlier Frames
PAM	Parameterized Appearance Model
PCA	Principal Component Analysis
PDF	Probability Density Function
PET	Positron Emission Tomography
POSIT	Pose from Orthography and Scaling by Iterations
SBA	Sparse Bundle Adjustment
SCM	Shape Conversion Matrix
SDM	Supervised Descent Method
SfM	Structure from Motion

SFS	Shape From Shading
SIFT	Scale-Invariant Feature Transform
SOP	Scaled Orthographic Projection
SSMR	Supervised Sparse Manifold Regression
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SVR	Support Vector Regressor
TAI	Tracking Accuracy Index
TMS	Transcranial Magnetic Stimulation
UPNA	Universidad Pública de Navarra
WCS	World Coordinate System
wPOSIT	Weighted Pose from Orthography and Scaling by Iterations

CHAPTER 1

Introduction

The aim of this introductory chapter is to present the framework in which the thesis is defined. Section 1.1 is focused on the motivations of the thesis through an exhaustive review of the state of the art of head pose estimation. General objectives of the thesis are covered in Section 1.2 and, finally, a summary of the structure of the work is provided in Section 1.3.

1.1. Motivation and State of the Art

People have always been able to provide and receive information from the movements of the head (i.e. position and orientation variations), which are a nonverbal form of communication that very often accompanies the speech. The interpretation of these head movements is crucial for understanding the intentions of other people in everyday human interactions, since the head pose gives direct information of people's attention target. In a computer vision framework, head pose estimation (HPE) is understood as the calculation of the head position and orientation with respect to the camera in front of which the person is located. This must be accomplished using the two-dimensional images obtained from that camera. Full orientation in 3D is determined by six degrees of freedom, namely the three rotation angles (*roll*, *yaw*, *pitch*) and the three translations (T_x, T_y, T_z) along the three spatial axes that define the World

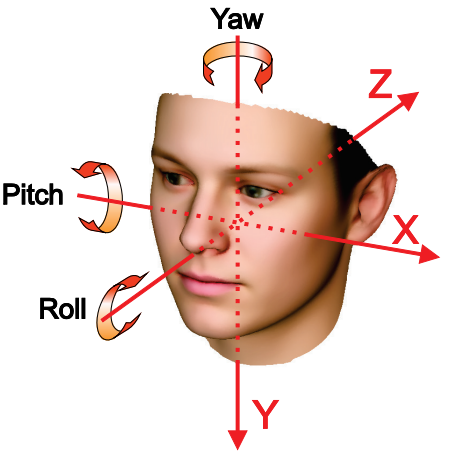

Figure 1.1. Definition of the spatial translations and rotations that determine the head pose.

Coordinate System (WCS) (see Fig. 1.1). HPE systems usually provide the transformation that needs to be applied to the head reference system in order to align it with the camera, which typically concurs with the WCS.

HPE offers a wide range of applications that covers human-computer interaction (HCI), human behavior analysis, driver assistance systems, pose-invariant face recognition, or several biomedical engineering tasks, amongst others. There is a great amount of information contained in head gestures, such as people facing each other when engaging discussions, nodding when they agree or understand, shifting the head towards a specific direction when they switch the target of attention and there is an object of interest around, and much more. HPE is a rich form of communication and, according to the scientific community, it will probably become another bridge for the interaction between humans and computers [1]. It is, for instance, a key point for different applications in the biomedical engineering framework. Long-duration nuclear-medicine brain-imaging techniques, such as positron emission tomography (PET), suffer from quality loss due to patient movement, and image correction can be achieved through HPE [2]. In transcranial magnetic stimulation (TMS), a very specific brain region is stimulated through a robotic arm in a non-invasive way, and inaccuracies in the area location may lead to serious consequences. HPE is often used to correct the arm position when the patient moves involuntarily [3].

HCI has experienced an important rise in the past decade due to its multidisciplinary nature and its application in a vast number of fields, such as assistive technologies, artificial intelligence or control of mobile devices. Lately, research on HCI has focused on developing control methods without the need of touch, such as hand gesture recognition [4], [5], head tracking [6]–[8], or gaze estimation [7], [9], [10], among others. HCI based on eye-control is rapidly evolving and has a great potential, mainly due to the enormous spread of mobile devices [11]. In fact, gaze tracking is currently one of the most popular HCI methods, with many applications in healthcare, mostly when neurological impairments are involved and it is not possible to ask for user cooperation. Gaze tracking systems suffer in unconstrained environments because they are very sensitive to head motion, and HPE has become a key point for successful gaze estimation [12]. Head movement compensation techniques, without knowing the head pose, have been largely studied with different success [13]–[15]. However, it has already been demonstrated that the head pose is as important as the eye location in order to determine the gaze direction [16], and gaze tracking and HPE are often combined in the search for better gaze estimation accuracy without constraining user movements, which also increases the application range [17]–[19]. Moreover, when the eyes are not visible for some reason, such as shadowing or people wearing glasses, the head pose gives a good estimate of the gaze direction. And, of course, when doing gaze estimation in low resolution images, in which the irises are not represented in great detail, it becomes of critical importance to have an accurate HPE system that provides additional and very valuable information.

These observations are nowadays widely accepted by the scientific community, but a long way has been walked along since the first observations of this problem were made: in 1824, the British chemist and natural philosopher William Hyde Wollaston showed what today is known as the Wollaston illusion [20]. In the drawing (Fig. 1.2), two heads are observed in different poses, and the gaze is perceived in different directions in spite of the eyes being drawn in exactly the same configuration in both images. If the heads were removed from the drawing, both pairs of eyes would be seen identical and gazing at an approximately frontal direction. This gave the first notion of the head pose strongly affecting the perception of the gaze direction. It is therefore evident that an eye tracker should be complemented by a head pose estimator in order to be able to reproduce the human gaze perception, assuming this one is an accurate gaze estimation system.

Figure 1.2. Wollaston illusion, drawn in 1824. The two heads are in different poses, and the gaze is perceived in different directions in spite of the eyes being drawn in exactly the same configuration in both images.

An ideal head pose estimator would provide results totally independent from any conditions altering the images, such as lighting changes, distortion, noise or facial expressions. This is usually very complicated to achieve, and that is why different approaches have been taken to the problem, each of them with their strengths and weaknesses, and therefore more or less suitable depending on the requirements of the application for which the system is intended. At one end, we find systems oriented to getting a coarse head pose estimate, usually from a discrete set of poses, sometimes just discerning a frontal pose from a non-frontal one, or a head moving to the left from a head moving to the right. The range of applications of this kind of systems may be more limited, but there are cases where a simpler head pose estimator like this satisfies the requirements and is the best suited option. At the other end, we find head pose estimators measuring a continuous pose in different degrees of freedom (DOF), typically three translations and three rotations. Moreover, applications in which not only the head pose is measured, but also different facial expressions are recognized and integrated in the system adding more DOFs have been significantly growing in the past few years.

As described in [1], the existing HPE methods can be classified based on the conceptual approach they use. Some of these approaches are usually oriented to getting a coarse head pose estimate, such as appearance templates or detector arrays. Other approaches provide a coarser or finer estimate depending on their specific implementation, such as nonlinear regression methods or manifold embedding methods. And some others are typically intended for a finer HPE, such as flexible models, geometric methods, tracking methods, or hybrid methods

that combine some of the former in an attempt to take advantage of the strengths of each.

Methods based on appearance templates usually try to find the best match between the input image and a set of templates corresponding to a discrete set of poses. These methods are usually simple to implement and more templates can be added at any point, incrementing the range of poses covered and adapting to changes in conditions. For example, Niyogi and Freeman [21] calculated the mean squared error over a sliding window in order to find the most similar template and assign its labeled pose to the current image. The main limitation of these systems comes from the differences in appearance that are not due to the head pose, such as physiognomic differences between users or facial expressions. Some attempts have been made to overcome these limitations, including some post-processing steps such as a convolution with a Gabor wavelet to emphasize the horizontal or vertical distribution of facial features from the nose or mouth [22].

Detector arrays are based on using a series of face detectors, each of them typically trained to detect a discrete pose. A classic example would be the method developed by Huang et al. [23], where they implemented three support vector machines (SVM) to distinguish three discrete yaws. Detector arrays can overcome the problem of appearance changes not due to the pose variation with an appropriate training, but they are not well suited for real time applications, precisely due to the amount of training necessary. In fact, they are usually limited to just one DOF and less than 12 detectors, which limits significantly their range of application. The larger the number of detectors is, the harder it is for a training sample to be positive in the corresponding detector and negative in all the rest. Some work was carried out to deal with this issue by Rowley et al. in [24], where they proposed using multiple networks with a *router* classifier determining the input pose first and then picking a single detector to support or discard the decision, but this technique was only tested for roll (in-plane) rotation.

Methods based on appearance templates or detector arrays are always intended for coarse HPE, thus meaning that their accuracy is never high and the error may often be above 10°, depending on the number of templates or detectors employed.

Non-linear regression methods consist in finding a functional mapping between features observed in the image and the corresponding head pose. Some approaches have tried to employ regression tools, such as support vector

regressors (SVR), to find a head pose from data of reduced dimensionality after applying principal component analysis (PCA) as in [25], localized gradient orientation histograms as in [26], or from previously detected facial features as in [27]. Han et al. recently proposed incorporating image abstraction to HPE [28], creating binary images with the most important features of the face. They then applied local directional quaternary patterns (LDQP) to get a better representation for pose classification. However, the most widely employed regression tool in the literature for HPE has been neural networks, in many different configurations. The most basic one consists in a multilayer perceptron (MLP), with as many hidden layers as desired, that is usually trained by a supervised learning technique that back-propagates the error through the cells to set the configuration of weights in the network. Some approaches have worked on having a discrete pose at each output node of the network [29]–[31], which leads to a coarse estimation of the pose, whereas other approaches have tried to achieve a finer estimation by having one DOF at each output node [32]–[34]. More recently, Zhang et al. combined a fuzzy c-means algorithm to extract clustering centers for previously detected facial points with neural networks to integrate HPE in a driver assistance system [35]. Another recent approach was proposed in [36], where Guo et al. employed tensor learning models for regression in a supervised learning framework, generalizing well-known regression schemes to tensor-based regression. Errors of non-linear regression methods usually range between 5-10°, and the implemented approaches are consequently intended for finer or coarser HPE depending on their accuracy.

Manifold embedding methods work by creating a low-dimensional continuous manifold structure from the high-dimensional image space. This manifold must be constrained by the possible head pose variations and an embedding technique is necessary in order to project new samples onto the manifold. The head pose can then be recovered for new face images by applying regression in the embedded space or using embedded template matching. Typical manifold learning techniques include locally linear embedding (LLE) [37], isometric feature mapping (Isomap) [38] and Laplacian eigenmaps (LE) [39]. In [40], Raytchev et al. described a nonlinear pose image expression technique for an Isomap manifold, in order to embed new data samples into the manifold. Techniques for embedding new samples for LLE and LE manifolds have been harder to achieve. In [41], Fu & Huang proposed embedding new samples by approximating LLE with locally embedded analysis (LEA). They built a neighborhood weight graph under LLE through graph embedded linearization and then found the projection direction. He et al. [42] proposed locality

preserving projections (LPP) as a linearized version of LE manifolds for visual recognition.

One of the main problems when working with manifold embedding methods is that they usually operate in an unsupervised fashion and therefore ignore the pose labels that are usually available during training. Therefore, manifolds are not built only for pose, but also for appearance variation (identity, lighting…), making it more difficult to recover the pose correctly for new image samples. Balasubramanian et al. [43], [44] proposed a solution based on biased manifold embedding (BME), in which they used a distance metric biased toward samples with smaller pose difference. Head pose label information was used before obtaining the low-dimensional embedding, and a generalized regression neural network was applied to learn the nonlinear mapping. This method showed better performance for HPE than Isomap, LLE or LE. Recently, Wang et al. [45] proposed a supervised sparse manifold regression (SSMR), in which they combined a supervised graph Laplacian regularization for better describing the dominant features while effectively preserving the local structure, and a sparse regression to recover the head pose. A similar approach was adopted in [46], where the supervised manifold learning was divided in three stages: neighborhood construction, graph weight computation and projection learning. HPE accuracy of these manifold embedding methods vary among approaches intended for finer of coarser estimation, but is usually in a similar range as non-linear regression methods (i.e. 5-10° of average error).

Methods based on flexible models are based on fitting nonrigid models to match the facial configuration of the subject in the image. The model is usually defined by a set of facial features that are iteratively deformed until the best match is achieved according to a previous training. The most widespread nonrigid model fitting techniques are active shape models (ASM) and active appearance models (AAM), techniques introduced by Cootes et al. [47]–[50] more than a decade ago and very popular today. These methods are based on a previous learning stage using a training image set in which key segmentation features have been annotated. ASM learns the statistical behavior of the object shape and the appearance of the neighborhood of each landmark, whereas AAM learns textures in the regions between landmarks in addition to the object shape. The training samples are images where the object has already been segmented. Segmentation consists in defining the shape of interest through the placement of several landmarks distributed along it, each of them defining a unique point of correspondence between all the shapes. Using these training samples, ASM and

AAM learn the patterns of variability of the object and are able to automatically locate the landmarks. While ASM only works with shape constraints, AAM models also the texture of the object, creating a combined model from the coupled relationship between shape and texture [51]. PCA is usually applied to the training data in order to reduce its dimensionality and obtain the modes that best describe the observed variation.

Once the statistical model has been generated from a training set, a fitting can be achieved for each new input image: a coarse estimate of the face location is typically used to initialize the algorithm, and it then iteratively adjusts the appearance parameters by comparing the rendered model and minimizing the distance to the image observation, until the rendered shape converges to the image feature locations. Simple approaches have tried to estimate the pose with a single DOF by mapping those parameters to, for example, a yaw estimate using linear regression [52]. More sophisticated approaches have worked on developing a combined 2D+3D AAM in order to determine the 2D facial feature locations and the 3D head pose simultaneously. In [53], Xiao et al. built the corresponding 3D shape modes of a 2D AAM by employing a nonrigid structure from motion (SfM) algorithm and then used these modes to constrain the 2D AAM fitting process and generate only instances observed by the 3D modes. SfM can also be used to directly estimate the 3D shape and the pose difference between frames in an image sequence [54]. Saragih et al. [55] adopted a deformable model fitting strategy in which they imposed joint motion constraints over facial landmarks in order to introduce regularization and avoid degenerate solutions. Very recently, Vicente et al. [56] applied a supervised descent method (SDM) for fitting a parameterized appearance model (PAM), which showed to be more efficient than usual fitting methods, and the head pose was obtained along the optimization process.

Methods based on flexible models have shown to be considerably accurate, usually reporting HPE errors between 3-5°. However, there is a lot of uncertainty about their performance in low resolution far-field imagery due to the difficulty in achieving good fitting and precise image feature location in those conditions. Besides, the sensitivity to the model initialization is always inherent to these methods and it is not easy to obtain appropriate model parameters to re-initialize the fitting when it has previously failed. In order to overcome these problems, Sung et al. [57] proposed combining AAMs with a cylindrical head model (CHM) and using the global pose parameters given by the latter to help the fitting and re-initialization of the statistical model. The idea consists in estimating

the pose by a CHM fitting algorithm, back-projecting the 2D AAM fitted points to the cylinder, transforming it to the calculated pose, and re-projecting them onto the image plane to obtain the AAM parameters as fitting initialization. They showed that combining the AAM with the CHM effectively strengthens the former in terms of pose estimation. As an alternative, An & Chung [58] modeled the face as a 3D ellipsoid and employed least-squares minimization to find motion parameters, assuming a first order approximation of the rotation to linearize the pose estimation problem. These methods have also reported HPE errors between 3-5° for low resolution far-field imagery.

Geometric methods are based on using the cues that human perception uses to estimate the head pose according to psychophysical studies, such as the location of the facial features with respect to the face contour, the orientation of the nose contour in the image plane, or that of the line of the mouth. These methods are simpler and significantly faster, but they usually provide worse HPE results and present more restrictions. One of their main disadvantages is the need of a very accurate detection of particular facial points, which is always hard to achieve and sometimes even impossible, mostly when working with low resolution far-field images. Differences among most approaches of this group rely on the set of facial points selected and the geometrical characteristics assumed for the 2D-3D relation that gives the head pose. As an example, Wang and Sung [59] automatically detected the four corners of the eyes and the corners of the mouth in the image, drawing three lines (inner corners of the eyes, outer corners, and corners of the mouth). The vanishing point of the three lines in the image plane was calculated using least squares and the 3D pose was calculated assuming the ratios between lines were known, applying a Gaussian mixture model to account for the differences between subjects and minimize the back-projection error. In [60], Gee and Cipolla used the outer corners of the eyes, the corners of the mouth and the tip of the nose, drawing a symmetry axis between midpoints of the eyes' line and mouth's line to calculate the 3D angle of the nose and determine the facial direction. In [61], Sun and Yin identified two clusters of inner eye corners from a 3D facial mesh model without texture and detected the nose tip with the aid of a facial reference plane, creating a symmetry plane and estimating the 3D pose from it. Dahmane et al. also exploited the bilateral face symmetry in order to estimate the head pose [62]. Recently, another approach [63] has been proposed in which the relationship between eyes' area and whole frontal face's area was used to obtain the pivot point in which head rotation is originated, and the relative position of certain facial features detected in the image with respect to the pivot point was used to calculate the head pose. Similarly, Hatem et al. [64]

calculated the head pose from the relative position of the eyes and mouth corners with respect to the nose. All these approaches are simple to implement, fast to run, and very useful in certain applications. Nevertheless, in general, it is recommended to combine this kind of techniques with other type of methods in order to obtain a more robust estimator.

Tracking methods rely on a prior knowledge of the starting pose of a sequence of images in which the head and its pose are to be tracked. Using that information, these methods are able to estimate the incremental head pose that accounts for the variation in the image observation, and typically show a high accuracy. Temporal continuity and smooth motion are usually assumed, and the estimator is often initialized from a frontal position with respect to the camera. Some basic approaches approximate the face by a planar surface and estimate the pose using least squares to find the best affine transformation that fits the image observation [65]. Recently, some works have started to use more complex feature descriptors to match points between frames. Scale-invariant feature transform (SIFT) [66] is currently the most employed descriptor that extracts distinctive invariant features from the image and performs a matching between different views of the object. These techniques have been used with the goal of accounting for large head movements, and combined with a RANSAC-based registration algorithm to detect outliers and estimate the pose [67]. Prasad and Aravind [68] combined SIFT matching with a parameterized face mask, the CANDIDE-3 model in its updated version [69], to fit the deformable model onto the observed image, also in a RANSAC configuration to avoid the outliers from SIFT, and applied the POSIT (Pose from Orthography and Scaling by Iterations) algorithm [70] to recover the head pose, reporting a high accuracy. Jang and Kanade [71] combined SIFT matching with a CHM and stored those features dynamically at certain reference poses to strengthen the pose estimation against occlusions and point drifting. Other approaches use this kind of feature matching just as an initialization to later refine the tracking with more accurate techniques. In [72], Malciu and Prêteux applied a simple block matching to initialize the pose estimation and account for large displacements, then using optical flow and an adjusted ellipsoidal model in an iterative configuration of 2D/3D matching to reach an optimal pose estimate.

Some efforts have also been made to overcome problems derived from illumination changes during tracking, which sometimes causes effects such as partial shadowing. A similarity metric learned from edge-density features extracted from training images can be used to handle strong lighting changes [73].

Using a CHM and its texture map image, the registration error in varying illumination conditions can be modeled as a linear combination of texture warping templates and orthogonal illumination templates, and minimized applying weighted least squares to achieve an stable tracking of the head [74].

One of the main problems of tracking-based approaches is the accumulated error that may appear as the appearance model updates, also known as *drifting*. In [75], Xiao et al. used templates of the head image and the corresponding head pose that were dynamically updated along the tracking, which was performed using iteratively re-weighted least squares (IRLS). When a new image was found to be in a pose close to that in one of the templates, the image was re-registered to the template to minimize error accumulation. Particle filters based on the appearance of the head have also been largely used in the literature to improve the tracking by maximizing the posterior probability using a motion history model. Once the head dynamics have previously been modeled, virtual views of the head can be rendered and compared with the current observation in order to update the weights of the particle filter. Several works have employed this kind of approach for tracking-based HPE [76]–[79]. Kalman filters have also been introduced in tracking-based approaches, trying to use information from predicted head poses in order to improve the tracking itself [80]. Recently, Wang et al. [81] developed a method based on keypoint matching, learning keypoint descriptors invariant to different viewpoints, nonrigid deformations and illumination changes. Color information was used in order to eliminate keypoints lying outside the face, and optical flow applied in order to remove motion jitter. Even more recently, Tran et al. [82] trained their method using synthetic data generated using a parametric 3D model and manually annotated facial landmarks, applying SIFT descriptors to learn an appearance model. Tracking was then performed and the head pose estimated using posterior probability, updating the tracking model through SVM. This work was further developed in [83], adding recovery capacity when tracking is lost. These tracking-based HPE methods are among the most accurate and have typically reported average errors between 2-4°.

Finally, hybrid methods are those that result from different combinations of the techniques described above. The basic idea behind hybrid implementations is to take advantage of the strengths of the methods that are combined and use them to overcome the limitations of each other, leading to a more accurate and robust head pose estimator. A usual approach consists in applying a static estimator to initialize the pose and then pass this information to a faster and more accurate tracking-based method. Some static HPE methods are

not fast or accurate enough for a continuous tracking, but can provide a good enough initialization for one of the tracking-based methods, whose limitation is often the need of an initial pose estimate, or a reinitialization when drifting problems arise. PCA can be applied for appearance learning and template matching online, and optical flow constraints for 3D head motion estimation [84]. PCA embedded skin-tone and edge-based detection was combined with a multi-state continuous density hidden Markov model in [85], and PCA template matching with stereo tracking to get view-based appearance models in [86], what was called the adaptive view-based appearance model (AVAM). This work was further developed by the same authors in [87], where they introduced the generalized AVAM (GAVAM) model, where three estimators were combined: a differential tracker using the previous and current frame, a keyframe-based approach using the current frame and a set of selected frames, and finally a static head pose estimator using just the current frame.

Point tracking methods and geometry-based methods are one of the most recurrent combinations in the literature. Early approaches include the one by Horprasert et al. [88], where five facial points were tracked across frames and symmetry of the eyes and anthropometric statistics were applied to recover the head pose. Hu et al. [89] obtained an initial estimate of the pose by analyzing the asymmetry of facial components in the image and then tracked the face using a geometrical model. Stereo-based tracking can also be combined with facial geometry, as in [90]. Recently, Chun & Kim [91] proposed an optical flow facial feature tracking framework with template matching with the initial features to avoid losing track, and created 3D vectors from the 2D points to geometrically estimate the head pose.

Hybrid methods can also be understood as those who apply different techniques for face tracking, getting an output from each of them to then transform these results in a single head pose estimate. These outputs may be complete head pose estimates or some intermediate cues. The key point of these methods resides in fusing the information in an intelligent way so that the final HPE becomes more robust or accurate, or both. An interesting approach in this direction was presented by Sherrah and Gong [92], where they fused similarity-to-prototype measures and skin colour information based on the estimation of covariance from observations. The fused information was introduced in terms of constraints to the head tracker to get the head pose. A more recent approach was presented by Valenti et al. [19], [93], where head pose and eye location information were combined to obtain an enhanced gaze estimation. The HPE

was carried out by implementing the tracking-based method presented by Xiao et al. [75]. Although the final goal was to achieve an accurate gaze direction estimate, the HPE was corrected in the process using the information obtained from the transformation matrix generated by the found eye location. Lefevre & Odobez [94] worked with two facial point sets, one randomly picked and the other one previously trained, and a deformable 3D model for both. An appearance model was trained for the second set, and tracking performed on both. Best HPE results were achieved by the hybrid method using both sets, where the adaptive features provided accuracy whereas trained features provided stability, both combined into a maximum likelihood framework. A very recent approach was presented by Asteriadis et al. [95], where they combined local features with appearance information, fusing them in a common framework. Local information flow was obtained using distance vector fields (DVF) as feature tracker [96], whereas holistic appearance information was obtained through the use of convolutional neural networks (CNNs).

Hybrid methods are currently the most popular ones because of their versatility and the possibility of improving the typical accuracy-robustness tradeoff. Current accuracy of these methods is also in the range of 2-4° depending on the image conditions.

1.2. Objectives

According to what has been said so far, the general purpose of this thesis is to propose new techniques that contribute to the development of head pose estimation methods, focusing the efforts on achieving improvements for techniques based on image feature tracking. In particular, the objectives of this work are the following:

❖ To summarize the state-of-the-art of the main HPE techniques employed so far in the literature. Mainly, this analysis has been focused on techniques oriented to getting a continuous head pose estimate over a sequence of frames.

❖ To build a framework for this thesis, consisting of different tools for the study of sources of error in HPE and for the evaluation of methods. This

framework will give us criteria to choose or discard different options for implementation.

❖ To develop and evaluate various face reconstruction techniques, establishing a reliable method to recover a 3D head model from 2D image observations.

❖ To implement an accurate and fully-automatic head pose estimator based on 2D facial feature tracking. The system has to be able to automatically detect and track facial points across an image sequence, build and fit a 3D head model based on the 2D observations, and provide a head pose estimate for every frame.

Those objectives describe the aim of this work regarding the specific task of head pose estimation. However, this thesis aims to achieve a wider impact by addressing the problem from a more general perspective. In this sense, the following additional objectives are defined:

❖ To design solutions to overcome the typical *drifting* problem present on general 2D feature tracking systems, consisting in the accumulation of error as the system loses track.

❖ To build a bridge between the 2D image and 3D object spaces, with the goal of considering mutual cues so that the 2D appearance is used to improve the 3D reconstruction and vice versa.

❖ To develop techniques for the specific task of head pose estimation that can be generalized to any kind of 3D object tracking.

By fulfilling the described objectives, this thesis tries to contribute to the development of accurate and robust head pose estimation methods, which are one of the most important trends in computer vision due to the huge proliferation of human-computer interfaces. It is important to note that the thesis

is oriented to developing techniques applicable not only to head pose estimation, but to generic object tracking.

1.3. Structure of the Thesis

This thesis is organized in five chapters, being the first one the present introductory chapter. It has been focused on explaining the motivations of this work, along with an overview of the state of the art of HPE techniques. Besides, the objectives of the thesis have been established. The following paragraphs provide a summary of the other four chapters of this work.

Chapter 2 presents a specific framework that has been developed to support the rest of the thesis. First, a head pose database of videos has been recorded in order to establish a reference for head pose estimation evaluation and replace out of date databases. Second, a simulation environment has been designed and implemented, which allows us to perform experiments in completely controlled conditions and study different parts of a head pose estimator independently. Finally, a generic 3D object pose estimation algorithm has been characterized using the simulation tools developed, in order to show its performance and study its suitability for the specific task of HPE. This chapter is particularly relevant for this thesis, as the techniques proposed in the next chapters are going to rely on this framework for drawing conclusions.

Chapter 3 covers the topic of 3D face reconstruction, designing and studying different model fitting methods with the aim of building an accurate head model based on 2D observations. The proposed methods will be tested on the simulation environment of Chapter 2, since it presents an important advantage: the real full 3D head model is available, which gives us a ground truth for comparison. Moreover, the simulator allows us to study the model fitting problem in an independent way, without head tracking issues affecting the results. By having the performance of the proposed reconstruction methods compared, the most suitable one to be integrated in a full head pose estimator can be chosen.

Chapter 4 is focused on the pose estimation topic. It is first addressed as a generic object pose estimation problem. The basic approach to generate a pose estimate from 2D observations and a reconstructed 3D model of the object is described, followed by some methods to improve the estimate. On the one hand, it is shown that the performance of the pose estimation algorithm can be

enhanced by introducing weights in the point correspondences. On the other hand, a method to calculate a 2D tracking accuracy index for each image point is proposed, through some novel invariant shape metrics. This index can then be used to obtain the weights to apply in the pose estimation algorithm. The method is then naturally extended to perform outlier detection. Moreover, an outlier correction technique is proposed to enhance both the 2D feature tracking and 3D pose estimation. The specific problem of HPE is then addressed. First, 2D facial point detection and tracking methods are proposed and evaluated. Then, all the improvements proposed before for generic pose estimation are adapted to HPE, describing the particularities and proposing specific solutions. To close the chapter, the proposed methods are evaluated for HPE using the head pose database of videos described in Chapter 2, as well as another widespread database that allows carrying out a comparison with other state of the art HPE methods.

Finally, Chapter 5 provides a compilation of the most significant results and general conclusions of this thesis. Besides, future research lines related to this work are proposed and briefly analyzed.

Bibliography of the Chapter

[1] E. Murphy-Chutorian, S. Member, and M. M. Trivedi, "Head Pose Estimation in Computer Vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, 2009.

[2] P. Bühler, U. Just, E. Will, J. Kotzerke, and J. Van Den Hoff, "An Accurate Method for Correction of Head Movement in PET," *IEEE Trans. Med. Imaging*, vol. 23, no. 8, pp. 1176–1185, 2004.

[3] L. Richter, L. Matthäus, and A. Schlaefer, "Fast robotic compensation of spontaneous head motion during Transcranial Magnetic Stimulation (TMS)," in *UKACC International Conference on Control*, 2010, pp. 1–6.

[4] K. Wang, L. Xu, Y. Fang, and J. Li, "One-against-all frame differences based hand detection for human and mobile interaction," *Neurocomputing*, vol. 120, pp. 185–191, Nov. 2013.

[5] U. Lee and J. Tanaka, "Finger Identification and Hand Gesture Recognition Techniques for Natural User Interface," in *APCHI'11: Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, 2013, pp. 274–279.

[6] Y. Qiao, X. Xie, T. Sun, and Y. Li, "A Design of Human-Computer Interaction Based on Head Tracker," *2008 IEEE Pacific-Asia Work. Comput. Intell. Ind. Appl.*, pp. 718–721, Dec. 2008.

[7] D. Zhu, T. Gedeon, and K. Taylor, "Exploring Camera Viewpoint Control Models for a Multi-Tasking Setting in Teleoperation," in *CHI'11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 53–62.

[8] K. Yamaguchi, T. Komuro, and M. Ishikawa, "PTZ Control with Head Tracking for Video Chat," in *CHI EA '09: CHI '09 Extended Abstracts on Human Factors in Computing Systems*, 2009, pp. 3919–3924.

[9] X. Yuan, Q. Zhao, D. Tu, and H. Shao, "A Novel Approach to Estimate Gaze Direction in Eye Gaze HCI System," *2013 5th Int. Conf. Intell. Human-Machine Syst. Cybern.*, vol. 588, pp. 588–591, Aug. 2013.

[10] J. Mansanet, A. Albiol, R. Paredes, J. M. Mossi, and A. Albiol, "Estimating Point of Regard with a Consumer Camera at a Distance," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7887 LNCS, pp. 881–888, 2013.

[11] S. Srinivas and J. Blustein, "A Survey Report on Mobile Eye-Based Human-Computer Interaction," *Int. Conf. Inf. Soc. i-Society*, pp. 146–151, 2011.

[12] J. Sigut and S. Sidha, "Iris Center Corneal Reflection Method for Gaze Tracking Using Visible Light," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 2, pp. 411–419, 2011.

[13] Z. Zhu and Q. Ji, "Novel Eye Gaze Tracking Techniques Under Natural Head Movement," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 12, pp. 2246–2260, 2007.

[14] C. A. Hennessey and P. D. Lawrence, "Improving the Accuracy and Reliability of Remote System-Calibration-Free Eye-Gaze Tracking," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 7, pp. 1891–1900, 2009.

[15] D. Cho and W. Kim, "Long-Range Gaze Tracking System for Large Movements," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 12, pp. 3432–3440, 2013.

[16] S. R. H. Langton, H. Honeyman, and E. Tessler, "The influence of head contour and nose angle on the perception of eye-gaze direction.," *Percept.*

Psychophys., vol. 66, no. 5, pp. 752–71, Jul. 2004.

[17] D. Cazzato, M. Leo, and C. Distante, "An investigation on the feasibility of uncalibrated and unconstrained gaze tracking for human assistive applications by using head pose estimation.," *Sensors (Basel).*, vol. 14, no. 5, pp. 8363–79, Jan. 2014.

[18] N. M. Bakker, B. A. J. Lenseigne, S. Schutte, E. B. M. Geukers, P. P. Jonker, F. C. T. Van Der Helm, and H. J. Simonsz, "Accurate Gaze Direction Measurements With Free Head Movement for Strabismus Angle Estimation," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 11, pp. 3028–3035, 2013.

[19] R. Valenti, N. Sebe, and T. Gevers, "Combining Head Pose and Eye Location Information for Gaze Estimation," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 802–815, 2012.

[20] W. H. Wollaston, "On the Apparent Direction of Eyes in a Portrait," *Philos. Trans. R. Soc. London*, vol. 114, pp. 247–256, 1824.

[21] S. Niyogi and W. T. Freeman, "Example-based head tracking," *Proc. Second Int. Conf. Autom. Face Gesture Recognit.*, pp. 374–378, 1996.

[22] J. Sherrah, S. Gong, and E. J. Ong, "Face distributions in similarity space under varying head pose," *Image Vis. Comput.*, vol. 19, no. 12, pp. 807–819, Oct. 2001.

[23] J. Huang, X. Shao, and H. Wechsler, "Face Pose Discrimination Using Support Vector Machines (SVM)," in *Proc. 14th Int'l conf. Pattern Recognition*, 1998, pp. 154–156.

[24] H. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 38–44.

[25] Y. Li, S. Gong, J. Sherrah, and H. Liddell, "Support Vector Machine Based Multi-View Face Detection and Recognition," *Image Vis. Comput.*, vol. 22, no. 5, pp. 413–427, 2004.

[26] E. Murphy-Chutorian, A. Doshi, and M. M. Trivedi, "Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation," in *2007 IEEE Intelligent Transportation Systems Conference*, 2007, pp. 709–714.

[27] H. Moon and M. Miller, "Estimating Facial Pose from a Sparse Representation," in *Proc. IEEE Int'l Conf. Image Processing*, 2004, pp. 75–78.

[28] B. Han, S. Lee, and H. S. Yang, "Head Pose Estimation using Image Abstraction and Local Directional Quaternary Patterns for Multiclass Classification," *Pattern Recognit. Lett.*, vol. 45, pp. 145–153, 2014.

[29] L. M. Brown and Y. L. Tian, "Comparative Study of Coarse Head Pose Estimation," in *Proc. IEEE Workshop Motion and Video Computing*, 2002, pp. 125–130.

[30] B. Schiele and A. Waibel, "Gaze Tracking Based on Face-Color," in *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 1995, pp. 344–349.

[31] L. Zhao, G. Pingali, and I. Carlbom, "Real-Time Head Orientation Estimation using Neural Networks," in *Proc. IEEE Int'l Conf. Image Processing*, 2002, pp. 297–300.

[32] E. Seemann, K. Nickel, and R. Stiefelhagen, "Head Pose Estimation using Stereo Vision for Human-Robot Interaction," in *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 2004, pp. 626–631.

[33] R. Stiefelhagen, J. Yang, and A. Waibel, "Modeling Focus of Attention for Meeting Indexing based on Multiple Cues," *IEEE Trans. Neural Networks*, vol. 13, no. 4, pp. 928–938, 2002.

[34] M. Voit, K. Nickel, and R. Stiefelhagen, "Neural Network-based Head Pose Estimation and Multi-View Fusion," in *Multimodal Technologies for Perception of Humans: Proc. First Int'l Workshop Classification of Events, Activities and Relationships*, 2007, pp. 291–298.

[35] L. Zhang, D. Zhang, Y. Su, and C. Wang, "Head Pose Estimation Based on Feature Extraction , Fuzzy C-Means and Neural Network for Driver Assistance System," in *IEEE Int'l Conf. on Control & Automation (ICCA)*, 2014, pp. 677–682.

[36] W. Guo, I. Kotsia, and I. Patras, "Tensor Learning for Regression," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 816–827, 2012.

[37] L. Saul and S. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, 2003.

[38] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A Global Geometric

Framework for Nonlinear Dimensionality Reduction," *Sci. 22*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[39] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[40] B. Raytchev, I. Yoda, and K. Sakaue, "Head Pose Estimation by Nonlinear Manifold Learning," in *IEEE International Conference on Pattern Recognition*, 2004, pp. 462–466.

[41] Y. Fu and T. S. Huang, "Graph Embedded Analysis for Head Pose Estimation," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, 2006, pp. 3–8.

[42] X. He, S. Yan, Y. Hu, and H.-J. Zhang, "Learning a Locality Preserving Subspace for Visual Recognition," in *Proc. IEEE Int'l Conf. Computer Vision*, 2003, pp. 385–392.

[43] V. N. Balasubramanian, J. Ye, and S. Panchanathan, "Biased Manifold Embedding: A Framework for Person-Independent Head Pose Estimation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[44] V. N. Balasubramanian, S. Krishna, and S. Panchanathan, "Person-Independent Head Pose Estimation Using Biased Manifold Embedding," *EURASIP J. Adv. Signal Process.*, pp. 1–15, 2008.

[45] Q. Wang, Y. Wu, Y. Shen, Y. Liu, and Y. Lei, "Supervised sparse manifold regression for head pose estimation in 3D space," *Signal Processing*, vol. 112, pp. 34–42, 2015.

[46] C. Wang and X. Song, "Robust Head Pose Estimation via Supervised Manifold Learning.," *Neural Netw.*, vol. 53, pp. 15–25, May 2014.

[47] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models-Their Training and Application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[48] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[49] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for

computer vision," 2004.

[50] I. Matthews and S. Baker, "Active Appearance Models Revisited," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 135–164, 2004.

[51] G. J. Edwards, a. Lanitis, C. J. Taylor, and T. F. Cootes, "Statistical models of face images — improving specificity," *Image Vis. Comput.*, vol. 16, no. 3, pp. 203–211, 1998.

[52] T. F. Cootes, K. Walker, and C. J. Taylor, "View-based active appearance models," *Proc. Fourth IEEE Int. Conf. Autom. Face Gesture Recognit. Cat No PR00580*, vol. 20, no. 9–10, pp. 657–664, 2000.

[53] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-Time Combined 2D+3D Active Appearance Models," *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004.*, vol. 2, 2004.

[54] Z. Gui and C. Zhang, "3D Head Pose Estimation using Non-Rigid Structure-from-Motion and Point Correspondence," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2006, pp. 1–3.

[55] J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable Model Fitting by Regularized Landmark Mean-Shift," *Int. J. Comput. Vis.*, vol. 91, no. 2, pp. 200–215, 2011.

[56] F. Vicente, Z. Huang, X. Xiong, F. De Torre, W. Zhang, and D. Levi, "Driver Gaze Tracking and Eyes Off the Road Detection System," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2014–2027, 2015.

[57] T. Kanade, J. Sung, and D. Kim, "Pose Robust Face Tracking by Combining Active Appearance Models and Cylinder Head Models," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 260–274, Jan. 2008.

[58] K. H. An and M. J. Chung, "3D Head Tracking and Pose-Robust 2D Texture Map-Based Face Recognition using a Simple Ellipsoid Model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2008, pp. 307–312.

[59] J. Wang and E. Sung, "EM enhancement of 3D head pose estimated by point at infinity," *Image Vis. Comput.*, vol. 25, no. 12, pp. 1864–1874, 2007.

[60] A. Gee and R. Cipolla, "Determining the Gaze of Faces in Images," *Image Vis. Comput.*, vol. 12, no. 10, pp. 639–647, 1994.

[61] Y. Sun and L. Yin, "Automatic Pose Estimation of 3D Facial Models," in *IEEE Conference on Pattern Recognition (ICPR)*, 2008.

[62] A. Dahmane, S. Larabi, I. M. Bilasco, and C. Djeraba, "Head Pose Estimation Based on Face Symmetry Analysis," *Signal, Image Video Process.*, Jul. 2014.

[63] M. S. L. Khan, S. U. Réhman, Z. Lv, and H. Li, "Head Orientation Modeling: Geometric Head Pose Estimation using Monocular Camera," in *Proc. of the 1st IEEE/IIAE Int'l Conf. on Intelligent Systems and Image Processing*, 2013, pp. 149–153.

[64] H. Hatem, Z. Beiji, R. Majeed, and J. Waleed, "Head Pose Estimation Based on Detecting Facial Features," *Int. J. Multimed. Ubiquitous Eng.*, vol. 10, no. 3, pp. 311–322, 2015.

[65] P. Yao, G. Evans, and A. Calway, "Using affine correspondence to estimate 3-D facial pose," in *Proc. IEEE Int'l Conf. Image Processing*, 2001, vol. 2, pp. 919–922.

[66] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[67] G. Zhao, L. Chen, J. Song, and G. Chen, "Large Head Movement Tracking Using SIFT-Based Registration," in *Proc. ACM Int'l Conf. Multimedia*, 2007, pp. 807–810.

[68] B. H. P. Prasad and R. Aravind, "A Robust Head Pose Estimation System for Uncalibrated Monocular Videos," in *Proc. 7th Indian Conf. Computer Vision, Graphics and Image Processing*, 2010, pp. 162–169.

[69] J. Ahlberg, "Candide-3. An updated parameterised face," *LiTH-ISY-R-2326*, vol. 1, pp. 1–16, 2001.

[70] D. F. DeMenthon and L. S. Davis, "Model-Based Object Pose in 25 Lines of Code," *Int. J. Comput. Vis.*, vol. 15, no. 1, pp. 123–141, 1995.

[71] J.-S. Jang and T. Kanade, "Robust 3D Head Tracking by Online Feature Registration," *IEEE Int. Conf. Autom. Face Gesture Recognit.*, pp. 1–6, 2008.

[72] M. Malciu and F. Prêteux, "A Robust Model-Based Approach for 3D Head Tracking in Video Sequences," in *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 2000, pp. 169–174.

[73] Y. Wu and K. Toyama, "Wide-Range, Person-and Illumination-Insensitive Head Orientation Estimation," in *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 2000, pp. 183–188.

[74] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: anapproach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, 2000.

[75] J. Xiao, T. Moriyama, T. Kanade, and J. F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *Proc. Fifth IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2002.

[76] F. Dornaika and F. Davoine, "Head and Facial Animation Tracking using Appearancee-Adaptive Models and Particle Filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, 2004, vol. 10, pp. 153–162.

[77] K. Oka, Y. Sato, Y. Nakanishi, and H. Koike, "Head Pose Estimation System Based on Particle Filtering with Adaptive Diffusion Control.," in *Proc. IAPR Conference on Machine Vision Applications*, 2005, pp. 586–589.

[78] S. Choi and D. Kim, "Robust head tracking using 3D ellipsoidal head model in particle filter," *Pattern Recognit.*, vol. 41, no. 9, pp. 2901–2915, Sep. 2008.

[79] J. Tu, T. Huang, and H. Tao, "Accurate Head Pose Tracking in Low Resolution Video," in *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 2006, pp. 573–578.

[80] W. Yu and L. Gang, "Head Pose Estimation Based on Head Tracking and the Kalman Filter," in *Int'l Conf. on Physics Science and Technology (ICPST)*, 2011, vol. 22, pp. 420–427.

[81] H. Wang, F. Davoine, V. Lepetit, C. Chaillou, and C. Pan, "3-D Head Tracking via Invariant Keypoint Learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 8, pp. 1113–1126, 2012.

[82] N. Tran, F. Ababsa, M. Charbit, and D. Petrovska-delacr, "3D Face Pose and Animation Tracking via Eigen-Decomposition based Bayesian Approach," *Adv. Vis. Comput. Lect. Notes Comput. Sci.*, vol. 8033, pp. 562–571, 2013.

[83] N.-T. Tran, F. Ababsa, and M. Charbit, "A Robust Framework for

Tracking Simultaneously Rigid and Non-rigid Face using Synthesized Data," *Pattern Recognit. Lett.*, vol. 65, pp. 75–80, 2015.

[84] Y. Z. Y. Zhu and K. Fujimura, "Head Pose Estimation for Driver Monitoring," in *Proc. IEEE Intelligent Vehicles Symposium*, 2004, pp. 501–506.

[85] K. S. Huang and M. M. Trivedi, "Robust Real-Time Detection , Tracking , and Pose Estimation of Faces in Video Streams," in *Proc. International Conference on Pattern Recognition (ICPR)*, 2004, pp. 965–968.

[86] L.-P. Morency, A. Rahimi, and T. Darrell, "Adaptive View-Based Appearance Models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 803–810.

[87] L. P. Morency, J. Whitehill, and J. Movellan, "Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation," in *IEEE International Conference on Automatic Face & Gesture Recognition*, 2008, pp. 1–8.

[88] T. Horprasert, Y. Yacoob, and L. S. Davis, "An Anthropometric Shape Model for Estimating Head Orientation," in *Proc. Third Int'l Workshop Visual Form*, 1997, pp. 247–256.

[89] Y. H. Y. Hu, L. C. L. Chen, Y. Z. Y. Zhou, and H. Z. H. Zhang, "Estimating Face Pose by Facial Asymmetry and Geometry," in *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 2004, pp. 651–656.

[90] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky, "Real-Time Stereo Tracking for Head Pose and Gaze Estimation," in *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 2000, pp. 122–128.

[91] J. Chun and W. Kim, "3D Face Pose Estimation by a Robust Real Time Tracking of Facial Features," *Multimed. Tools Appl.*, 2014.

[92] J. Sherrah and S. Gong, "Fusion of Perceptual Cues for Robust Tracking of Head Pose and Position," *Pattern Recognit.*, vol. 34, no. 8, pp. 1565–1572, 2001.

[93] R. Valenti, Z. Yucel, and T. Gevers, "Robustifying Eye Center Localization by Head Pose Cues," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2009, pp. 612–618.

[94] S. Lefèvre and J. M. Odobez, "Structure and Appearance Features for

Robust 3D Facial Actions Tracking," in *Proceedings IEEE International Conference on Multimedia and Expo, ICME*, 2009, pp. 298–301.

[95] S. Asteriadis, K. Karpouzis, and S. Kollias, "Visual Focus of Attention in Non-calibrated Environments using Gaze Estimation," *Int. J. Comput. Vis.*, vol. 107, no. 3, pp. 293–316, 2014.

[96] S. Asteriadis, N. Nikolaidis, and I. Pitas, "Facial Feature Detection using Distance Vector Fields," *Pattern Recognit.*, vol. 42, no. 7, pp. 1388–1398, 2009.

CHAPTER 2

Framework

This chapter presents various tools that have been developed during this thesis in order to build a framework that will be fundamental for the experimental part of the thesis, as it will be evident in the following chapters. In order to evaluate and compare different head pose estimation (HPE) systems, ground truth data is necessary. The term *ground truth* refers to the absolute truth of something and, in the context of this thesis, will refer to the real position and orientation of the head with respect to the camera when we speak of a *3D ground truth,* or the real image position of facial features when we speak of a *2D ground truth*. The 3D ground truth can be used to evaluate the HPE accuracy of the method, whereas the 2D ground truth is useful to assess the accuracy of facial feature detection and tracking across images. Moreover, methods based on a previous training usually require annotated faces in different poses. This annotation is usually carried out manually, becoming a tedious but necessary task to provide a good learning to the method.

Getting accurate ground truths and labeled faces is a very challenging task, and various efforts have been made in this chapter to address these issues. In Section 2.1, a new head pose database of videos is presented, the 'UPNA head pose database' [1]. To the best of our knowledge, no modern databases oriented to tracking-based HPE methods are available in the literature and, hence, there is an evident need for a new database, recorded in current technological conditions and with a thorough calibration process that allows a minimum-noise ground

truth. This database has been recorded with the aid of ten subjects and is public for research purposes.

There are many sources of noise when real videos are recorded to build a head pose database with real users. Since it is not possible to characterize each noise independently from the others using such a database, a simulation environment for HPE has also been developed in this thesis and is presented in Section 2.2. The idea consists in creating a totally controlled environment in which several types of head tracking and pose estimation experiments can be carried out and where each source of error can be manipulated at free will. This is a very useful tool that allows us to study the impact of different type of errors in HPE. This part has concluded with the creation of a synthetic head pose database, consisting of ten virtual users mimicking the head movements contained in the database described in Section 2.1. This synthetic head pose database is also public for research purposes.

Finally, a characterization of the basic pose estimation algorithm employed in this thesis, the POSIT algorithm, is presented in Section 2.3. POSIT is a generic object pose estimation method based on projection geometry and it is one of the fundamentals on which this work stands. As such, a preliminary analysis of its sensitivity to a variety of HPE noise sources is carried out in this section in order to study its feasibility for the task proposed in this thesis and set a starting point for accuracy in HPE. The study has been carried out in the controlled simulation environment described in Section 2.2.

2.1 UPNA Head Pose Database [1]

HPE methods based on a continuous head tracking and pose estimation need databases of videos with ground truth data for every frame in order to assess their performance. The most widespread database for this purpose is the Boston University Headtracking Database (BU database) [2]. This database was presented in the year 2000, and consists of 45 video sequences of 5 different subjects under uniform illumination conditions recorded with a Sony Handy-cam on a tripod at 30 frames per second. Each image sequence is 200 frames long and the frame size is 320×240 pixels. Each subject appears in 9 videos, in which they carry out free movements combining translations and rotations along the three spatial axes. Head rotation ranges go up to ±30 degrees, and head pose ground truth is provided for each video. This ground truth was collected using a "Flock of Birds"

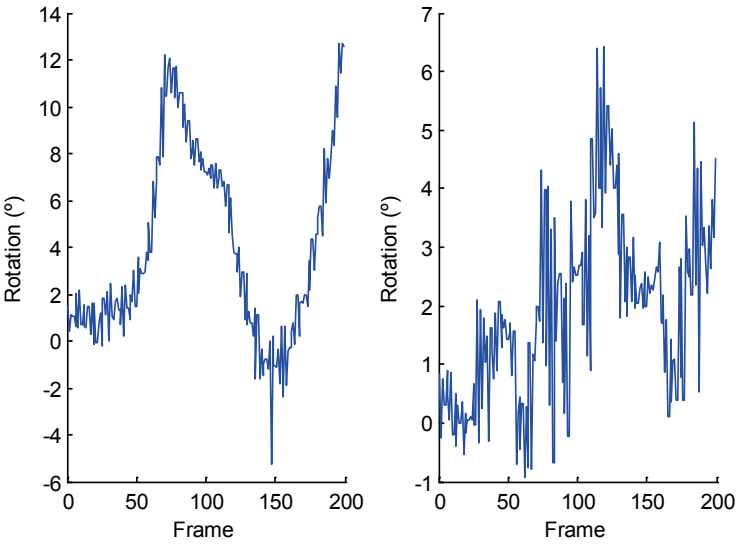

Figure 2.1: Two example ground truths of rotation around one of the axes taken from the BU Headtracking Database. The noise is visually perceptible in both curves, producing peaks and sharp variations of a few degrees during the recording.

magnetic tracker system [3], with the transmitter attached to the user's head. The nominal accuracy of the magnetic device was of 0.1 inches for position and 0.5 degrees for rotation. However, La Cascia et al. reported in their article [2] a lower accuracy due to metal devices interfering with the transmitted data. This noise is visually perceptible if we represent the ground truth against time for any video of the database, as shown in Fig. 2.1. Another inconvenient of this database is the unavailability of camera calibration data, such as focal length, optical center or distortion characterization.

In this thesis, a new database of videos for head pose estimation is presented with the goal of establishing a new framework for HPE algorithm validation, replacing out of date frameworks. It incorporates an automatic facial annotation system, designed also as part of this thesis, that enables us to label each video frame. A commercial magnetic sensor-transmitter system has been employed in order to obtain the head pose ground truth. There are several sources of noise during the recording process, e.g. nominal accuracy of the magnetic head tracker, slight movements of the headband with the sensor worn by the user, estimation of the *camera* – *transmitter* transformation matrix, or

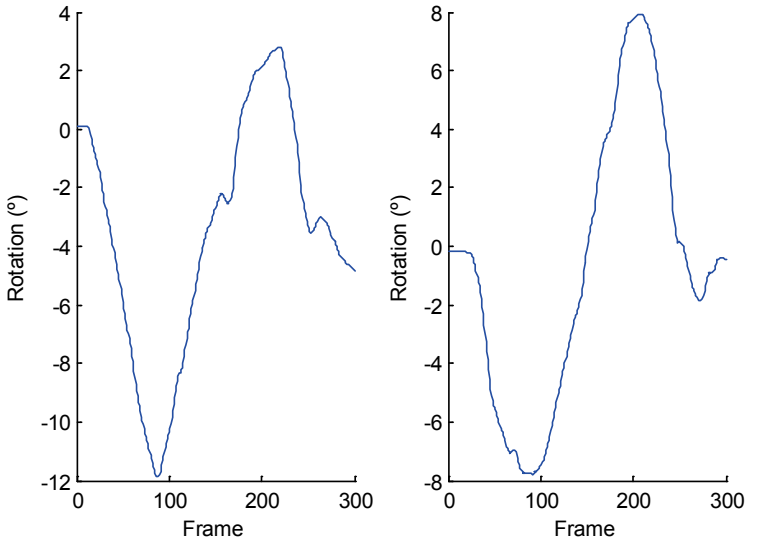

Figure 2.2: Two example ground truths of rotation around one of the axes taken from the UPNA Head Pose Database that has been built in this thesis. The noise is significantly lower in our ground truth than in BU's ground truth, as it can be observed by comparing this image with Fig. 2.1.

annotation of faces in 3D, and every measure has been taken in order to minimize these errors. However, it is not possible to get completely free of them and head tracking and pose estimation performed on this database will show an estimation error that will contain, to some extent, the noise produced by the sources described above.

This database presents three main contributions. First, significantly higher resolution and image quality can be obtained with current commercial web cameras compared to the BU database published in 2000. The use of a webcam is justified by the will of establishing a typical resolution and image quality in commercial user platforms, such as desktop and laptop computers, tablets, and other mobile devices.

Second, we have carefully designed a calibration method to align the camera and the transmitter reference systems, thus providing very reliable head pose ground truth and calibration parameters. Thanks to the thorough calibration, the noise in the head pose ground truth of our database is significantly lower compared to the BU database (see Fig. 2.2), which will allow

more reliable error estimations in this thesis as well as for future HPE works. Besides, all calibration parameters are available with our database (camera, pose data acquisition, and transformation matrices between camera – transmitter – sensor).

The third important contribution is that we have designed an automatic facial annotation method that allows us to provide 2D ground truth data containing the image position of several facial landmarks in every frame, thus avoiding the need of tedious manual annotation. Labeled datasets of training images are required for many applications, such as statistical approaches for building non-rigid deformable models. Active Shape Models (ASM) [4], Active Appearance Models (AAM) [5], [6] and 3D Morphable Models (3DMM) [7] are the most well-known examples of this family, and they all rely on the annotation of dozens or hundreds of corresponding landmarks across training images. When the annotation is done manually, in addition to the temporal cost of the task, the fact that the face is mainly characterized by edge structures makes it difficult to annotate the same landmarks consistently. In the case of the database presented here, as it will be better described in following sections, there are 36.000 frames in total, adding up all the users and videos. The task of manually annotating one face, following the 54-point model that will be presented later, takes 5 minutes in average to a trained observer. The annotation of the complete database would thus take 375 workdays for one person, which shows the value of the designed automatic annotation method.

The following subsections describe the database in further detail, paying special attention to the *camera – transmitter – sensor* system calibration and the automatic facial feature annotation procedure. A characterization of the error both in 2D and 3D introduced by these procedures is also presented.

2.1.1 General Description

The database consists of a set of 120 videos, corresponding to 10 different subjects (6 males and 4 females) and 12 videos each. Every set of 12 videos is composed of 6 guided-movement sequences and 6 free-movement sequences. In the guided sequences, the user follows a specific pattern of movement: three pure translations (X, Y, Z) and three pure rotations (roll, yaw, pitch). In the free sequences, the user moves the head at free will combining translations and rotations along the three spatial axes. The order of the videos is as follows:

Figure 2.3: Example frames from four different subjects in the database.

Video 1:	pure translation along the X axis
Video 2:	pure translation along the Y axis
Video 3:	pure translation along the Z axis
Video 4:	pure rotation around the Z axis (roll)
Video 5:	pure rotation around the Y axis (yaw)
Video 6:	pure rotation around the X axis (pitch)
Videos 7-12:	free rotations and translations

In order to provide the database with more uniformity, it has been considered convenient that every video begins and ends with the head in a frontal position, at a working distance from the camera (55-60cm). Movement ranges include translations going up to more than 200mm in any axis from the starting point, and rotations up to 30°. Fig. 2.3 shows four example frames taken from the database, with different subjects in different poses.

2.1.2 Hardware

The videos have been acquired using a commercial web camera, a Logitech HD Pro C920. It allows a maximum resolution, without interpolation, of 1920×1080 for static image acquisition, and of 1280×720 for video acquisition at 30 frames per second.

The 3D head position and orientation has been obtained using a "Flock of Birds" real-time motion tracking system [3], the 3D Guidance TrakSTAR

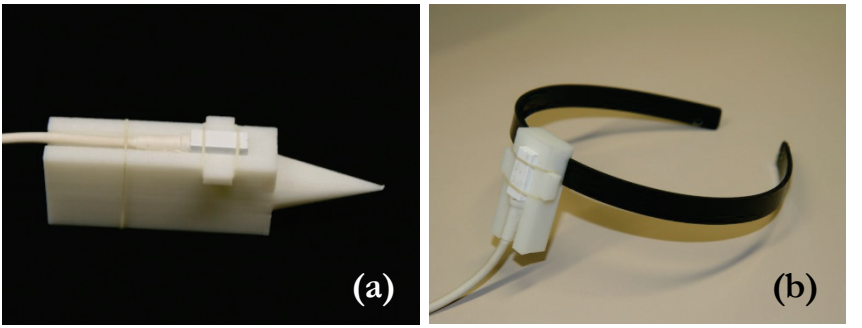

Figure 2.4: 3D-printed plastic pieces for accurate sensor handling. (a) Marker with a blunt point end to annotate facial landmarks in 3D, with a compartment for the sensor. (b) Piece with a groove to be attached to the headband and with a compartment for the sensor.

model from Ascension Technology Corporation. It consists of a transmitter that tracks the position of up to four magnetic sensors at a rate of 240Hz. Two of the sensors have been used for building the database: one is attached to the user's head during the video recording, whereas the other one is used to annotate the 3D position of the facial landmarks before the recordings, as explained in Section 2.1.5.

Two rigid plastic pieces have been designed and printed in 3D for handling the sensors with high precision (see Fig. 2.4). Both have a specific compartment with the exact dimensions of the sensor, so this can fit in and minimize undesired fluctuations that could affect the accuracy of the measurements. One is attached to a plastic headband that is adjusted to the user's head, whereas the other piece, with a blunt point end, is used as a 3D marker to annotate facial landmarks.

Controlled illumination for the recordings has been achieved using three quartz halogen lights and two diffusers.

2.1.3 Calibration Procedure

The whole system needs to be calibrated in order to obtain the position and orientation of the sensors with respect to the camera at every moment. The flock of birds gives their position with respect to the transmitter, which has a reference system totally independent from the camera's reference system. We need to know the transformation between the camera and the transmitter

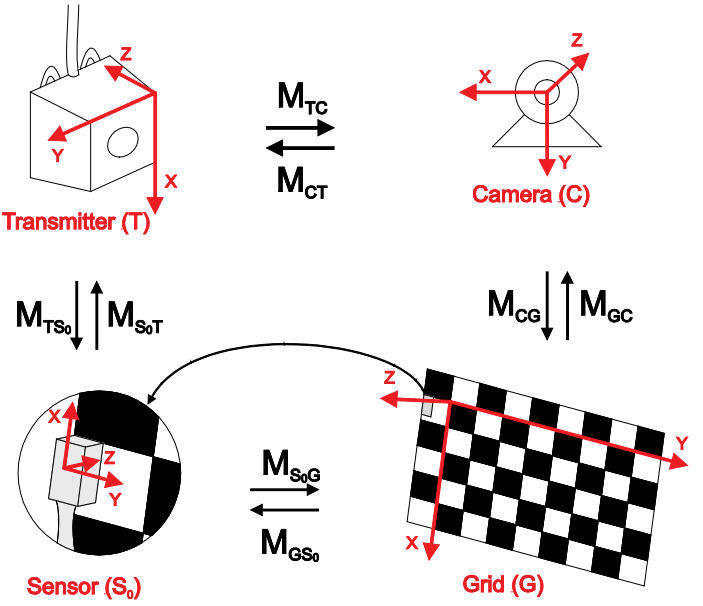

Figure 2.5: Complete calibration system. A calibration grid (G) with a sensor (S_0) attached to it in a specific location is used as an intermediate step that allows us to establish a final relationship between the transmitter (T) and the camera (C). Transformation matrices that relate the different elements are also represented.

reference systems in order to have a correspondence between the acquired images and the head pose recorded by the transmitter-sensor. In addition to this, we also need to synchronize in time both devices, so that we know what sensor data corresponds to what frame acquired by the webcam. This is achieved by transforming both to absolute times with a unique reference, and interpolating the sensor data to calculate the pose that corresponds to the exact timestamp of the camera.

Fig. 2.5 shows the complete calibration system where the different elements are indicated with letters for clarity in notation. Each element has its own reference system, and **M**'s are the transformation matrices between systems. As an example, $\mathbf{M_{TC}}$ refers to the 3D position and orientation of the transmitter (T) with respect to the camera (C). Transformation matrices are 4×4 and of the form:

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & t \\ 0 & 1 \end{bmatrix}$$

where $\mathbf{R}$ is the 3×3 rotation matrix, t the 3×1 translation vector, and the last row completes a square matrix in homogenous coordinates.

There is no way of establishing a direct path between the camera and the transmitter, so a calibration grid with a sensor attached to it is used as an intermediate step (elements G and S_0 in Fig. 2.5). The grid contains 28 squares of 30×30mm. Taking pictures of the grid in different positions and orientations with respect to the camera, intrinsic and extrinsic camera parameters can be accurately determined using the Matlab *Camera Calibration Toolbox* [8]. Intrinsic parameters (i.e. focal length, principal point, lens distortion coefficients) characterize the camera according to its fabrication, whereas extrinsic parameters make reference to the spatial position and orientation of the grid (G) with respect to the camera (C) for each of the pictures. These parameters can be used to determine the transformation matrices $\mathbf{M_{CG}^i}$ and $\mathbf{M_{GC}^i}$ between the camera and the calibration grid. We use the superscript *i* to show there are multiple positions of the grid and to make reference to each one of them. Attaching the sensor to the grid in a specific and known position and orientation, and knowing the reference systems of both elements, transformation matrices $\mathbf{M_{S_0G}}$ and $\mathbf{M_{GS_0}}$ between them can also be obtained, which are independent from the specific images. Furthermore, by acquiring data of the sensor position and orientation with respect to the transmitter, transformation matrices $\mathbf{M_{TS_0}^i}$ and $\mathbf{M_{S_0T}^i}$ are determined for each image. Using all the matrices above, the path *transmitter – sensor – calibration grid – camera* is built, and the matrices $\mathbf{M_{TC}}$ and $\mathbf{M_{CT}}$ for the direct transformation between the camera and the transmitter are calculated as follows (see Fig. 2.5):

$$\mathrm{M_{TC}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{M_{GC}^i} \cdot \mathrm{M_{S_0G}} \cdot \mathrm{M_{TS_0}^i}$$

$$\mathrm{M_{CT}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{M_{S_0T}^i} \cdot \mathrm{M_{GS_0}} \cdot \mathrm{M_{CG}^i}$$

where N is the total number of grid positions recorded, and thus the total number of images used in the calibration.

Each of the described stages introduces some amount of error in the calibration process. Calibration images have been obtained at a resolution of

1920×1080 pixels. The inaccuracy in the estimation of the pair ($\mathbf{M_{CG}^{i}}, \mathbf{M_{GC}^{i}}$) can be minimized by controlling the illumination during the image acquisition and obtaining pictures of the best possible quality, with sufficient position and orientation variety of the calibration grid. The number of images used for calibration is directly related to this last factor. The stability of the camera's intrinsic parameters with respect to the number of calibration images has been studied, showing that they tend to stabilize from 30 images on [9]. The inaccuracy in the estimation of the pair ($\mathbf{M_{TS_0}^{i}}, \mathbf{M_{S_0T}^{i}}$) can in turn be minimized by reducing the distance between the sensor and the transmitter as much as possible, as well as by isolating both elements from any metallic device. For this purpose, the transmitter has been hung from the ceiling using a non-metallic structure and placing it above the calibration grid.

Taking all the above into account, it is assumed that the main calibration inaccuracy is due to the ($\mathbf{M_{S_0G}}, \mathbf{M_{GS_0}}$) pair estimation. This transformation matrix has been theoretically determined, since the sensor is attached to the calibration grid in an a priori known position and orientation. This is a critical step, since the placement of the sensor is manually done on a space drawn next to the grid, thus resulting in possible inaccuracies on the theoretically determined matrices. In order to minimize this error, an optimization process has been performed [9]. The idea of this process is as follows: knowing the structure of the grid and the pair ($\mathbf{M_{S_0G}}, \mathbf{M_{GS_0}}$), the 3D points that define the corners of the grid can be projected onto the image plane using the estimated $\mathbf{M_{TC}}$ and the intrinsic parameters of the camera. The real location of these points in the image is in turn obtained using the automatic corner detection from the camera calibration toolbox. The 2D projection error can therefore be calculated at every step of the optimization process. This process will iteratively search for the rotation and translation in ($\mathbf{M_{S_0G}}, \mathbf{M_{GS_0}}$) that minimize the sum of projection errors for all the calibration images. An estimation of the resulting error will be presented later.

2.1.4 Recording Setup

Once the setup has been calibrated, the recording of the videos is accomplished. Fig. 2.6 shows a layout of the recording setup, in which illumination elements are shown, as well as camera, transmitter and user position arrangement. A classical diffuse front illumination has been used, with a back light for background filling. The user is sit in front of the camera wearing a headband that has a sensor, named S_1, attached to it.

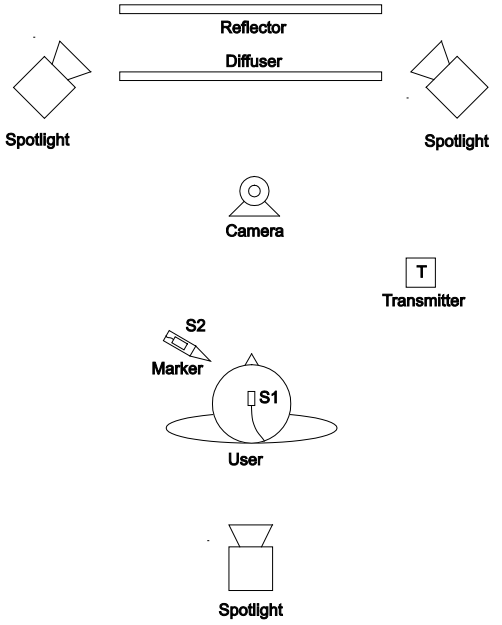

Figure 2.6: Recording setup showing all the elements involved in the recording of the videos. S_1 is attached to the user with a headband and S_2 is mounted on a marker for the 3D facial annotation.

The users have been asked to begin every video sequence in a frontal position and end it in the same position. They have had the aid of acoustic signals giving them timing information, so they can finish the corresponding movement and go back to the initial position just before the recording ends.

2.1.5 Automatic Facial Annotation

We propose a model of 54 facial landmarks to characterize the face, distributed along the eyebrows, eyelids, nose, lips, and jaw (see Fig. 2.7). In order to avoid the considerable effort of manually annotating the landmarks in each frame, we have designed an automatic annotation procedure, based on subject-specific 3D facial annotation and a subsequent projection onto the image plane. For this task, we use two sensors (S_1 and S_2), each of them with its corresponding reference system, independent from that of the calibration sensor S_0.

Since the sensor attached to the headband (S_1 in Fig. 2.8) moves jointly with the head, if we knew the coordinates of the facial landmarks in the reference

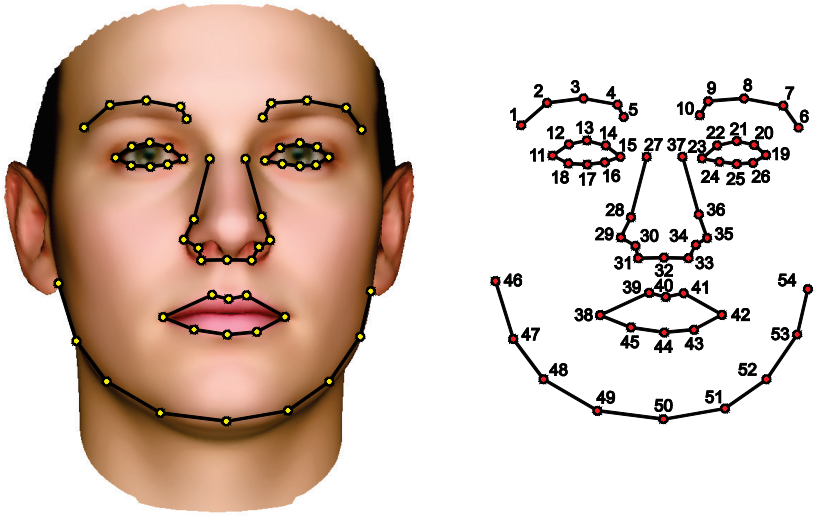

Figure 2.7: Model of 54 facial landmarks automatically annotated for the presented database. Anatomical placement (left) and numerical order (right) of the landmarks are shown.

system of that sensor S_1, we could project the facial features onto each frame according to the transformation matrices calculated in Section 2.1.3. For this purpose, the second sensor (S_2) has been used. The TrakSTAR device gives us the position and orientation of both sensors with respect to the transmitter ($\mathbf{M_{S_1T}}$ and $\mathbf{M_{S_2T}}$); thus, we can obtain the position and orientation of S_2 with respect to S_1 with an easy transformation, as shown in Fig. 2.8. $\mathbf{M_{S_2S_1}}$ represents the coordinates of S_2 in the reference system of S_1, and is calculated as:

$$M_{S_2S_1} = M_{TS_1} \cdot M_{S_2T}$$

S_2 is mounted on the plastic marker designed for the annotation task, as described in Section 2.1.2. Once the headband with S_1 has been firmly attached to the user's head, the 54 facial landmarks are annotated in 3D. The user is asked to keep a neutral expression during the annotation process as well as during the recording, for the 2D annotations to be as accurate as possible. The marker is held on each landmark for one second, so that 240 samples are recorded and averaged to minimize the inaccuracy due to the inevitable unsteadiness of the

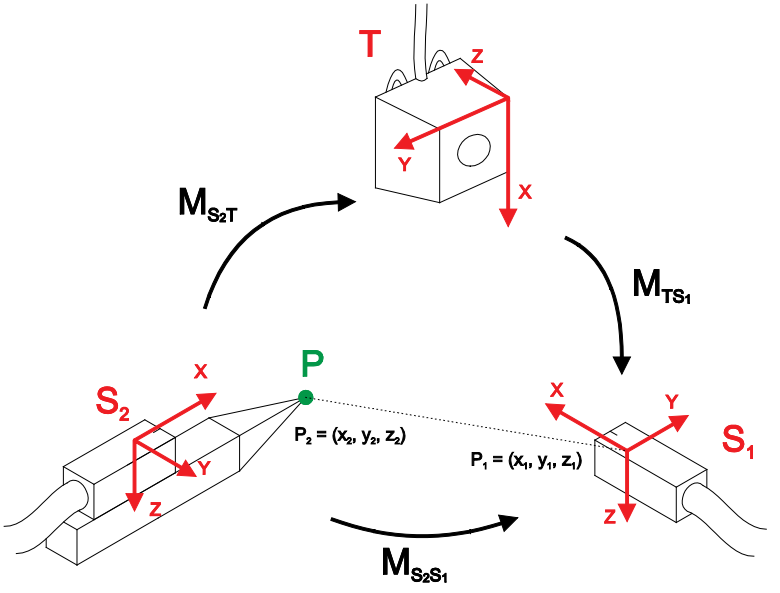

Figure 2.8: Diagram of the 3D facial annotation system. Sensor S_1 is placed on a headband attached to the head. Sensor S_2 is mounted on a marker. The marker's point is placed on the facial landmarks and S_2 records their position with respect to the transmitter (T). With an easy transformation, we can obtain their position with respect to S_1, given by $M_{S_2S_1}$.

hand. A picture showing a moment of this annotation process is shown in Fig. 2.9.

Since the marker dimensions are known, the tip of the marker is known and constant with respect to S_2. However, due to small tolerances in the fabrication process of the marker, there is some uncertainty in the tip's location. In order to eliminate it, a simple optimization has been performed. Given a fixed spatial point with respect to the transmitter, P in Fig. 2.8, the tip of the marker is placed on it and the marker is moved in all directions while the tip stays still, recording sensor-transmitter data for a few seconds. Every point of the marker is in constant movement in S_2's reference system except for the tip during this recording. Thus, the optimization looks for the coordinates of a point P with respect to the moving sensor for which the variability with respect to the transmitter during the sequence is minimal.

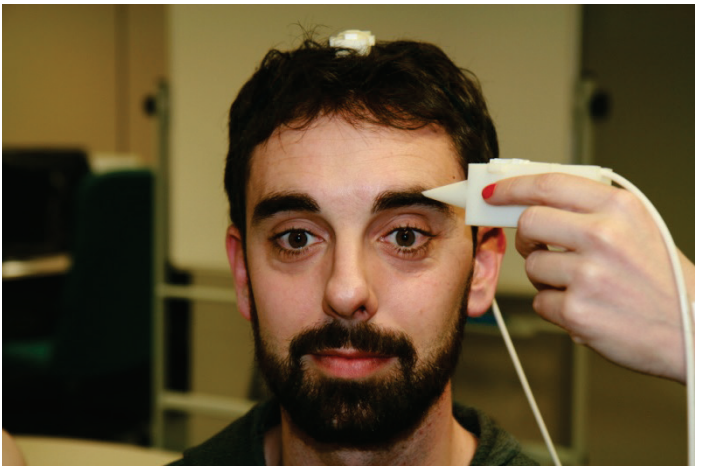

Figure 2.9: Picture showing a moment of the process of annotation of the 54 3D facial landmarks, using the fabricated marker with sensor S_2 mounted on it. The user is wearing the headband with sensor S_1 mounted on it as well.

Having annotated the 54 facial landmarks in 3D, we can use the transformation above ($\mathbf{M_{S_2S_1}}$) to obtain the coordinates of the 3D facial points in the reference system of S_1. Assuming S_1 stays still with respect to the head during the recording (the headband is used for that purpose), the location of the 3D facial landmarks with respect to S_1 is known at all times and they can be projected onto the image according to $\mathbf{M_{TC}}$, as shown in Fig. 2.10. Furthermore, since there may be slight movements of the headband with respect to the head during the recording process, another 3D facial landmark annotation has been performed at the end of the session of each user. The final 3D point set for each video has been calculated as an optimal linear combination of the initial and final annotated faces.

2.1.6 Structure and Content

As it has already been said, the database is publicly available for research purposes. It contains 120 videos corresponding to 10 different subjects, as mentioned above. The videos are provided in MPEG-4 format, recoded with a loss of approximately 1% with respect to the original recording. This way, the complete database occupies 200MB, instead of the 100GB that took up the original one, making it more manageable for researchers that want to download it.

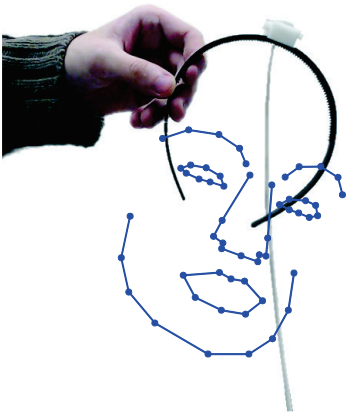

Figure 2.10: 54 facial points projected onto the image. They have previously been annotated in 3D while the user was wearing the headband. Thus the location of the 3D facial landmarks with respect to S_1 is known at all times and they can be projected in any pose.

The videos have a resolution of 1280×720 pixels, and have been acquired at 30 frames per second. Every video is 10 seconds long, containing 300 frames. Each video is associated with three ground truth text files. One contains the 2D projections (in pixels) of the annotated 3D facial points, what we will call the 2D ground truth. The other two contain the head pose with respect to the camera or what we will call the 3D ground truth. Translations are given in millimeters and rotations in degrees. The difference between the two 3D ground truths is that one contains the originally acquired head pose, whereas the other one contains the equivalent ground truth beginning with '0-rotation'. Basically, it is the original 3D ground truth transformed to get an exact zero rotation for the three angles in the first frame, assuming that the user is placed perfectly frontal with respect to the camera at the beginning of each video. This transformation is done by multiplying the inverse rotation matrix of the initial pose to the pose of each frame. Getting an exact zero initial rotation is not feasible during the recordings, in the original 3D ground truth, and applying this small transformation to every video is equivalent to moving the headband slightly at the beginning of each video so that the sensor gives an exact zero rotation for the initial frame. This transformation has been applied because every video ideally begins with a frontal face (see Section 2.1.1), and this assumption may be

important in some applications, where it may affect the estimation results, so it has been considered interesting to provide the zeroed 3D ground truth together with the original one. The deviation from the zero-rotation of the original 3D ground truth can be numerically measured as the mean absolute initial rotation. It is of 0.83°, 0.86° and 1.05° for roll, yaw and pitch respectively, which makes 0.91° in average.

The real 3D model of the subject annotated for each video is also provided in the database. The 3D facial points are referenced to the sensor S_1's origin.

The database has been organized by user. The video files, 2D and 3D ground truth files, and 3D model files of each user are contained in a separate folder. The database is publicly available for research purposes at http://gi4e.unavarra.es/databases/hpdb/. A leaderboard has also been created in the webpage so that HPE algorithms tested with the UPNA database can be sorted according to their results. Researchers can upload a text file with their head pose estimation for the complete database. They will automatically get their results against the 3D ground truth and see their position with respect to algorithms registered by other researchers. If they want their method to appear in the leaderboard permanently, they can submit the results and register the algorithm.

Table 2.1: Comparison between the UPNA database and the BU database.

Features	Our database	BU database
No. of users	10	5
No. of videos per user	12	9
No. of frames per video	300	200
Video resolution	1280×720	320×240
Type of movements	Free/Guided	Free
Camera calibration parameters	Provided	Not provided
3D head models	Provided	Not provided
3D ground truth	Provided	Provided
2D ground truth	Provided	Not provided
Leaderboard	Yes	No

The database's most important features are summarized in Table 2.1, where a comparison against the BU database [16] is carried out, since the latter has been a reference for evaluation of head pose estimation methods in videos for the last 15 years.

2.1.7 Error Characterization

The error in the automatic facial annotation has been characterized both in 2D and 3D [9]. The 2D projection error has been estimated using the calibration grid. The 16 corners of the grid have been annotated in 3D with the marker described in Section 2.1.2. 50 images of the calibration grid in different poses have been acquired with a resolution of 1920×1080 pixels, as in a typical calibration procedure. The 3D corners are then projected onto each image as in Section 2.1.5, and this projection is compared against the corners automatically detected in the image by the Matlab *Camera Calibration Toolbox*. The projection error has been calculated as the mean Euclidean distance between the projected point-cloud and the detected one.

This projection has been performed before and after the optimization of the pair ($\mathbf{M_{S_0G}}$, $\mathbf{M_{GS_0}}$), described at the end of Section 2.1.3, in order to observe the accuracy improvement given by this optimization during the calibration process. The mean projection error for the 50 images was of 8.88 pixels before the optimization and of 1.31 pixels after the optimization; therefore, the optimization process reduces the error in more than 85% of its value.

The 3D error of the annotation procedure has also been measured. A mannequin head with a realistic face appearance has been used for this purpose. On the one hand, the mannequin face has been annotated in 3D following the procedure described in Section 2.1.5. On the other hand, the whole head has been scanned with a 3D laser scanner, resulting in a dense point cloud [10]. After a manual pre-registration of both point sets, the 54 landmark set has been registered to the dense point cloud using an Iterative Closest Point (ICP) algorithm. The distance between both point clouds after registration has then been measured. For each landmark, the closest point in the dense cloud has been found, and the resulting distances have been averaged. This average distance between the annotated face and the scanned head represents the 3D error of the annotation procedure, and has been estimated in 0.71mm.

2.2 Simulation Environment

Traditionally, HPE methods have been trained or evaluated using images or video sequences of real people performing a variety of head movements. The problem with those environments is that it is impossible to control all the variables affecting the image and pose data acquisition or the labeling process for training. We have designed a simulation environment that allows us to create virtual images or videos for HPE, in which all those variables are controlled by the user and can be set in different manners depending on the goal of the study. The main advantage of this virtual environment is that it can be made completely noise-free, or with the exact amount of noise desired for each variable. Moreover, new video sequences can be created at any moment if new requirements are considered for the application and new studies want to be carried out, without the need of real subjects for the videos and the tedious task of setting up a new recording session.

This section is divided in two subsections. First, the simulator tool is described, specifying the control parameters that allow the user to define and build a specific sequence for HPE. Second, a synthetic database of videos for HPE is presented, which has been created with the simulator tool described next.

2.2.1 Simulator Tool

This tool has been fully developed in Matlab and can be divided in three main modules: the design of the simulation, where parameters that characterize the different variables of the simulation are specified according to the desired output; the building of the simulation, where the previously defined parameters are used to generate the output as specified; and, finally, a head pose estimation module where the generated simulation can be processed with the POSIT algorithm (see Section 2.3 for algorithm description) and the results are obtained. These modules have been designed to be run sequentially and will be described in detail in the following lines.

2.2.1.1 Simulation Designing

In this first step, the whole simulation is defined by setting different parameters that determine the output according to the needs of the user. The modifiable variables are the following:

Figure 2.11: The Basel Face Model (BFM). The mean shape and texture of the PCA are represented.

➢ **Head model:** the simulator incorporates a generative 3D shape and texture model, the Basel Face Model (BFM) [11]. It is a publicly available 3D morphable face model that can generate face images at any pose and under any illumination. The model was built based on training data obtained from the 3D scans of 200 subjects, 100 females and 100 males between 8 and 62 years old, most of them Caucasian. All the scans contained a neutral facial expression and were registered using an Optimal Step Nonrigid ICP Algorithm [12] to ensure optimized anatomical point correspondences between faces. The faces were parameterized as triangular meshes after registration, resulting in 53.490 vertices described by a coordinate vector $(x_i, y_i, z_i)^T \in \mathbb{R}^3$ with an associated colour $(r_i, g_i, b_i)^T \in [0,1]^3$. Principal component analysis (PCA) was then applied to create an orthonormal basis of 199 principal components of texture and shape, which allows us to generate new observations as linear combinations of those components. It is thus a face generator in which, just by assigning the PCA coefficients for the principal components of shape and texture, we can create new faces at any moment. Note that, if all coefficients are set to 0, the mean face of the PCA is obtained (Fig. 2.11).

The simulator is thus able to create new users from a *meta-database* of 3D faces. Besides, a certain set of facial points can be passed to the simulator so that their projection on the image plane is stored. Since there is an accurate anatomical correspondence between the vertices of the generated faces, those facial points can be stored as vertex indexes and will correspond to the same anatomical

location in any generated face. This allows us to obtain a 2D ground truth for any facial point set of interest for any video sequence generated with the simulator, which is very useful for training purposes or for point tracking algorithm evaluation.

- **Camera parameters:** we can set most of the parameters that define the image or video that would produce a real camera: the image resolution, the frame rate and the intrinsic camera parameters (i.e. focal length, principal point, radial and tangential distortion, and skew). The images or videos the simulator will produce will correspond to what a camera of those characteristics would acquire.

- **Motion:** the motion parameters define the head movements that will compose the created video. We can define the length of the sequence as a number of frames (we have already defined the frame rate), and the head movements in the 6 DOF (i.e. translation in X, Y, Z; roll, yaw, and pitch) that will determine the head pose in each of the frames.

- **Occlusion handling:** the defined ranges of movement (translation and rotation) and facial point set of interest determine the occlusions that will take place during the image sequence. Certain poses result in occluded parts of the face and this will be calculated for each frame during the building part described next. When a 3D facial point is occluded, the simulator offers three ways of handling it: 1) the real 2D projection of the point on the image plane is stored in the output even if it is not visible; 2) the closest visible point in the model is obtained and projected on the image, storing this 2D point in the output; 3) no projection is stored in the output. Option 2 is an approximation of a behavior often observed in real tracking: as the point starts to get occluded, the tracking drifts to another point in the neighborhood with a similar appearance, which we are approximating as the closest visible point in 3D. An example of this is shown in Fig. 2.12a, in which four points turn out to be occluded and their approximation by the closest visible point in the model is also shown. Summarizing, in the simulation designing stage we choose the way occluded points are going to be handled during the building.

- **Noise:** we can add noise to the 2D projections of the model points in order to simulate a non-ideal tracking. We apply a normally distributed random noise of mean zero to each 2D point in two possible ways: 1) same variance is used for all the points to generate the noisy observations; 2) the variance of the random noise applied to each point P is proportional to the angle formed by the vector normal to the surface of the 3D model just where P lies, and the vector

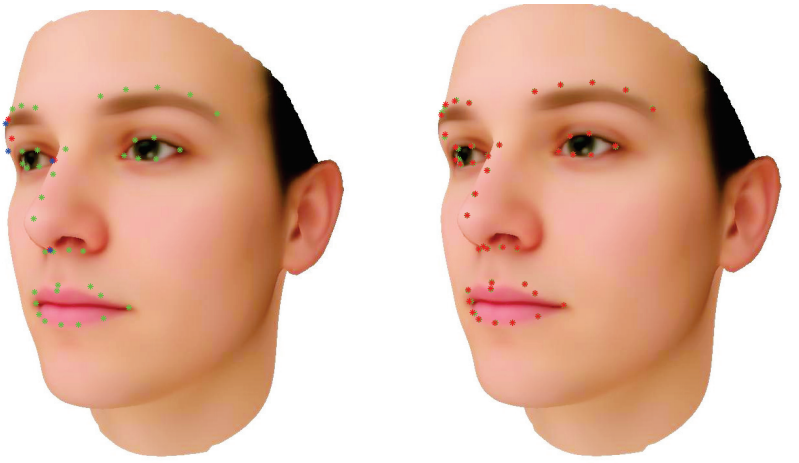

Figure 2.12: Examples representing the occlusion handling and tracking noise. (a) Occlusion handling: visible points (green), occluded ideal projections (red) and approximation by the closest visible point in the model (blue). (b) Tracking noise: ideal projections (red) and noisy projections (green).

that goes from the camera to P. The model is previously transformed to the pose that corresponds to the current frame. Option 2 means that facial points lying on parts of the face that "point" towards the camera will suffer from less tracking noise than those that lie on parts that point elsewhere. An of this second option example image is shown in Fig. 2.12b, where both the noisy and noise-free 2D points are represented. This is actually a similar approach to the one taken for the occlusion handling; instead of having a binary output (occluded/not occluded) and adding noise when the point is occluded (by approximating the point by the closest visible one), the noise is gradually incremented as the normal of the surface moves away from the line of sight of the camera and gets closer to an occlusion. If no noise is added, the ideal projections are obtained.

- **Output mode:** we can define whether we want a complete video, with the shape and texture model projected for each frame, to be created, or we just want to store the projection of the point set of interest without creating an actual video, which makes the building significantly faster. We can also set the compression format for the video if we choose to create it.

2.2.1.2 Simulation Building

This step consists in running the simulation according to the parameters defined in the previous step. The head model is generated based on the model parameters, and transformed in each frame with the corresponding rotation and translation.

The simulator also calculates for each video frame which part of the model is visible and which is not, applying the Hidden Point Removal (HPR) algorithm [13]. It consists in transforming a point cloud according to the viewpoint and extracting the points that reside on the convex hull, which leads to determining the visible points in the cloud. Each of the points of interest previously indicated to the simulator is then flagged as 'occluded' or 'not occluded' for each frame. When a point is occluded, the closest visible point of the model is obtained if this option has been set on in the simulation design.

The model is then projected onto the image plane using the shape and texture information in the corresponding pose. The point set of interest indicated in the previous step is stored separately, and it goes through a post-processing stage after projection. First, occlusions are handled. Occluded points are managed according to the option selected in the previous step. Points that go out of the field of view are also eliminated. After that, 2D noise is added to the point set in case it has been specified so in the simulation design.

Using all the previous information, the output of the simulation is generated: a video file is created (if specified) and all the parameters that define the simulation and the point set of interest are stored in *mat* files.

2.2.1.3 Head Pose Estimation

The simulator tool also incorporates a head pose estimation module based on the POSIT algorithm for generic object pose estimation described in Section 2.3. The main advantage of having this part also integrated in the simulator is that the processing of the simulation output is straightforward: 2D image points (noisy or not depending on the specifications) and 3D model points are passed onto the algorithm to get a head pose estimate, and HPE error is calculated using the ground truth specified in the design.

2.2.2 Synthetic Head Pose Database [14]

Using the tool described above, a complete database of videos for HPE has been created. This database is publicly available for research purposes in our group's webpage, i.e. http://gi4e.unavarra.es/. The basic idea behind this has been to recreate the database presented in Section 2.1, taking advantage of the simulator tool and the possibility to control every variable in play.

Similarly to the UPNA Head Pose Database, the synthetic database of videos consists of 120 videos of 10 different users. The first 5 users have been created using the head generator described in Section 2.2.1.1, assigning random coefficients to the principal axes in the PCA basis of the BFM. A large set of heads have been created using random shape and texture PCA coefficients from a Gaussian distribution with mean zero and standard deviation one, from which the 5 users have been visually selected in order to include a certain variety among the faces regarding the gender, size, age, and facial appearance in general. The last 5 users in the database have been selected from the 10 example scans provided along with the BFM in the webpage [15]. These scans have also been registered but have not been included in the training set of the morphable model, and are not therefore exactly reproducible by a certain set of PCA coefficients.

With the aim of having realistic head movements recreated in the synthetic head pose database, the ground truth of the UPNA Head Pose Database has been used. The synthetic heads are thus reproducing the exact pose variations of the real users. This allows us to assure certain realism in the synthetic database; rotations and translations at a constant speed were originally tried, and the visual effect was that of a robotic movement. Moreover, by copying the ground truth of the real database, we assure the same translation and rotation ranges are represented, and the resultant synthetic database will be of a similar difficulty for HPE algorithm testing.

12 videos per user have been thus generated, which include 6 guided-movement sequences and 6 free-movement sequences. The videos have been generated with a 1280×720 pixel resolution, at 30 frames per second, and stored in MPEG-4 format. Four example frames from the database are shown in Fig. 2.13. If we compare these example frames with the ones from the UPNA head pose database (Fig. 2.3), we can observe that the heads have similar proportions in the image, the appearance of the synthetic faces resembles reasonably that of a real face, and the main difference resides on the background, inexistent for the synthetic database.

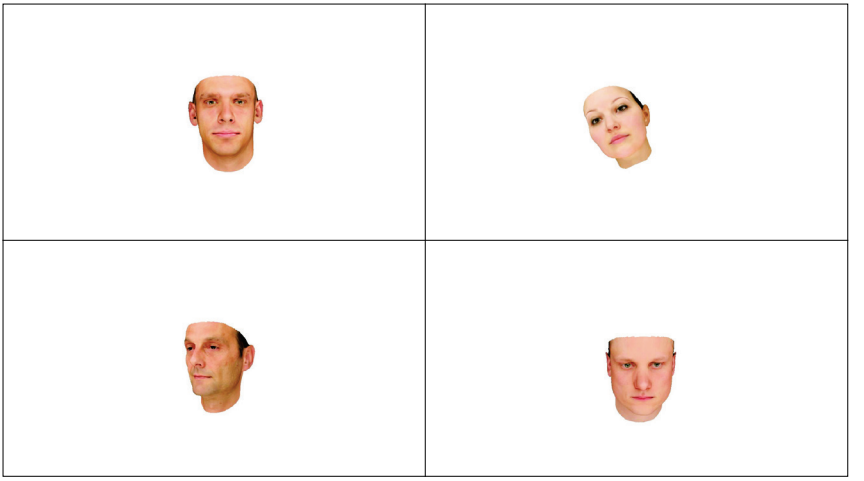

Figure 2.13: Example frames from four different subjects of the synthetic database.

A set of 43 facial points of interest has been defined as shown in Fig. 2.14, distributed along the eyebrows, eyelids, nose and mouth. These 3D landmarks have been annotated in the BFM and then projected onto each frame of the database and stored as the 2D ground truth.

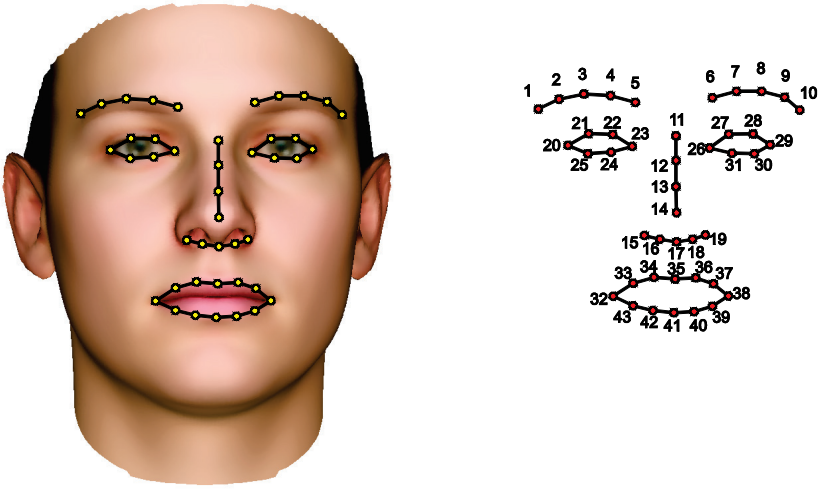

Figure 2.14: The 43-point model used for face annotation in the synthetic database.

As a result, the synthetic database provides the 120 videos with their corresponding 3D ground truth for HPE, 2D ground truth with ideal noise-free projections, the complete 3D head model of each user, and ideal camera parameters with which the videos have been generated.

2.3 POSIT Algorithm

POSIT stands for 'pose from orthography and scaling with iterations'. The algorithm was published by Daniel F. DeMenthon and Larry S. Davis in 1995 [16] as a generic object pose estimation method that could be written in 25 lines of code using Mathematica software. It is now a widely used method for estimating 3D object poses in space from 2D image observations. A translation vector t and a rotation matrix **R** are obtained from the algorithm, and they determine the transformation from the head coordinate system to the camera coordinate system. Roll, yaw, and pitch angles can then be computed from **R**.

This method has been considered a priori fit for the purpose of this thesis: it is fast, there are public implementations in C++ or Matlab among others, and it estimates the pose of any object from 2D images without the need of an initial pose estimate. This subsection first describes the algorithm in more detail for a better understanding of the pose estimation problem solving, and then presents a thorough experimental study that has been carried out in order to characterize the algorithm performance in different conditions. This study will help us determine its suitability for the specific task laid out in this work.

2.3.1 Algorithm Description

The POSIT algorithm can find the pose of any object from a set of at least four non-coplanar point correspondences between a 3D model of the object and a 2D image. It is thus assumed that at least four points of the object can be detected in the image and matched to their corresponding 3D points in the model, whose relative geometry is supposed to be known. The method is based on assuming a weak perspective model for an initial pose estimation, in which the points from the object are projected onto the image plane as scaled orthographic projections (SOP). Therefore, the weak perspective model is an approximation of the "true" perspective projection that considers all the model points to be on the same plane with respect to the camera (see Fig. 2.15).

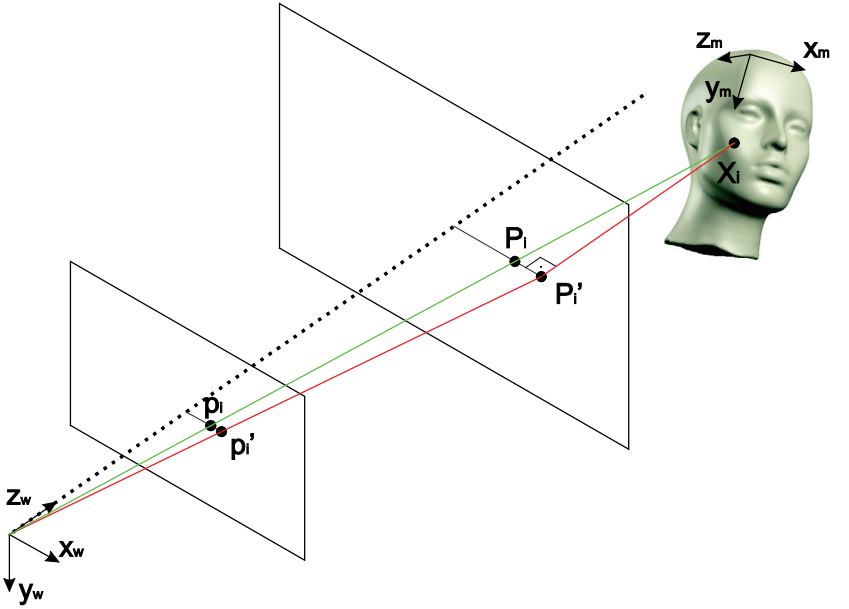

Figure 2.15: Illustration of the weak perspective model. Perspective projection (p_i) and scaled orthographic projection (p_i') of a 3D head point X_i. The camera or "world" reference system (x_w,y_w,z_w) and the object reference system (x_m,y_m,z_m) are also represented.

POSIT consists of two steps: the first one is defined as the pose from orthography and scaling (POS), an algorithm that uses the weak perspective projection to build a linear equation system whose solution leads to a rotation matrix and a translation vector representing the object pose that accounts for the image observation. This step is followed by an iterative process (IT) in which a scale factor for each point is calculated so that the originally found SOP is enhanced, meaning that it resembles the perspective projection in the best possible way. This is achieved by shifting the points of the object, in the pose calculated in the previous iteration, to the lines of sight, which is where they would be if the pose was correct. POS is then applied again to the corrected projections at each iteration, until a condition is satisfied: either a maximum number of iterations is reached, or convergence is achieved (i.e. the 2D difference between the previous and the current projections, averaged across all 2D correspondences, is smaller than a threshold).

2.3.2 Algorithm Characterization

The aim of the study that is presented in this section is to analyze the suitability of the POSIT algorithm for the specific task of HPE. For this purpose, we have decided to carry out three independent tests that show the performance of the algorithm against three parameters: the number of 2D-3D point correspondences, the image resolution, and the distance from the head to the camera. These parameters have been studied because they can be chosen by the user when defining the setup for HPE, and will usually depend on the target application. All of them have been studied under perfect 2D tracking conditions and assuming an ideal 3D model with exact 2D-3D correspondences is known. This is the way of getting a HPE error that will be intrinsic to the POSIT algorithm and just dependent on the application setup. How model inaccuracies affect POSIT will be studied in Chapter 3, and how tracking inaccuracies do, in Chapter 4.

Getting back to the setup parameters, the image resolution directly depends on the camera, which in turn depends on the application and, sometimes, on the characteristics of the device that integrates it. There will definitely be differences between a small camera integrated in a smartphone or tablet, an independent camera connected to a powerful desktop computer, or a driver assistance camera, for instance. Exactly the same reasoning can be applied to the distance from the head to the camera. The number of 2D-3D correspondences, however, usually depends on the feature detection and tracking algorithm. Some methods use fixed point meshes, whereas others adapt the number of points to the imaging conditions. In any case, from the state of the art review presented in the previous chapter, we can assume that a number between four or five and a few tens of facial features make up the typical range.

2.3.2.1 Number of 2D-3D Point Correspondences

This study has been carried out using the synthetic head pose database presented in Section 2.2.2. Based on the 43-point facial model defined in Fig. 2.14, ten different subsets of points have been defined, ranging from 4 to 43 points, as shown in Fig. 2.16. Starting from 4 points (outer eye corners and mouth corners), which is the minimum required by POSIT, more points have been gradually added until completing the 43-point shape. The criteria followed to define the subsets have been to keep as much symmetry as possible and always have most part of the face represented. The four points lying on the nasal septum have been added the last because those are the ones that break the coplanarity

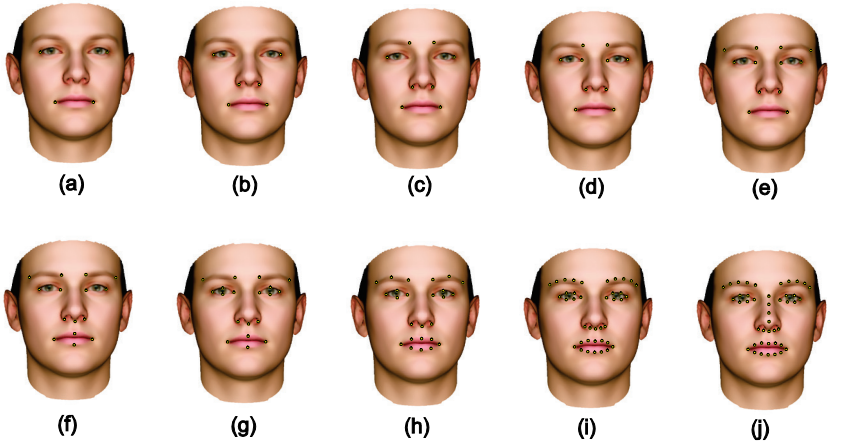

Figure 2.16: Different subsets of points defined for the study of the influence of the number of 2D-3D correspondences in POSIT. a) 4 points; b) 6 points; c) 8 points; d) 10 points; e) 12 points; f) 15 points; g) 19 points; h) 25 points; i) 39 points; j) 43 points.

more, which may be beneficial for POSIT according to the previous algorithm description.

Experimental results are shown in Fig. 2.17. As it could be expected from an intuitive point of view, the more point correspondences provided, the more accurate POSIT is. The case of using just 4 points is special and that is why it is not shown in the chart. Using just those points, the algorithm does not converge, and it is thus not able to provide a pose estimate, in 68% of the frames of the database. Moreover, when it does converge, the pose estimate is completely senseless in most cases, showing an average error of 211mm in translation and 41º in rotation. It is then evident that the outer corners of the eyes and corners of the mouth are not an appropriate set of points for running POSIT, either because the number of correspondences is too little, or the points are too coplanar in 3D to achieve accurate and converging pose estimates. For the rest of the cases, translation accuracy is in the range of tenths of millimeters, whereas rotation accuracy is in the range of hundreths of degrees. There is a significant improvement from using the 6-point model to the 8-point one, and the accuracy then increases more smoothly as more features are added to the model. The four points on the nasal septum, even though they are the least

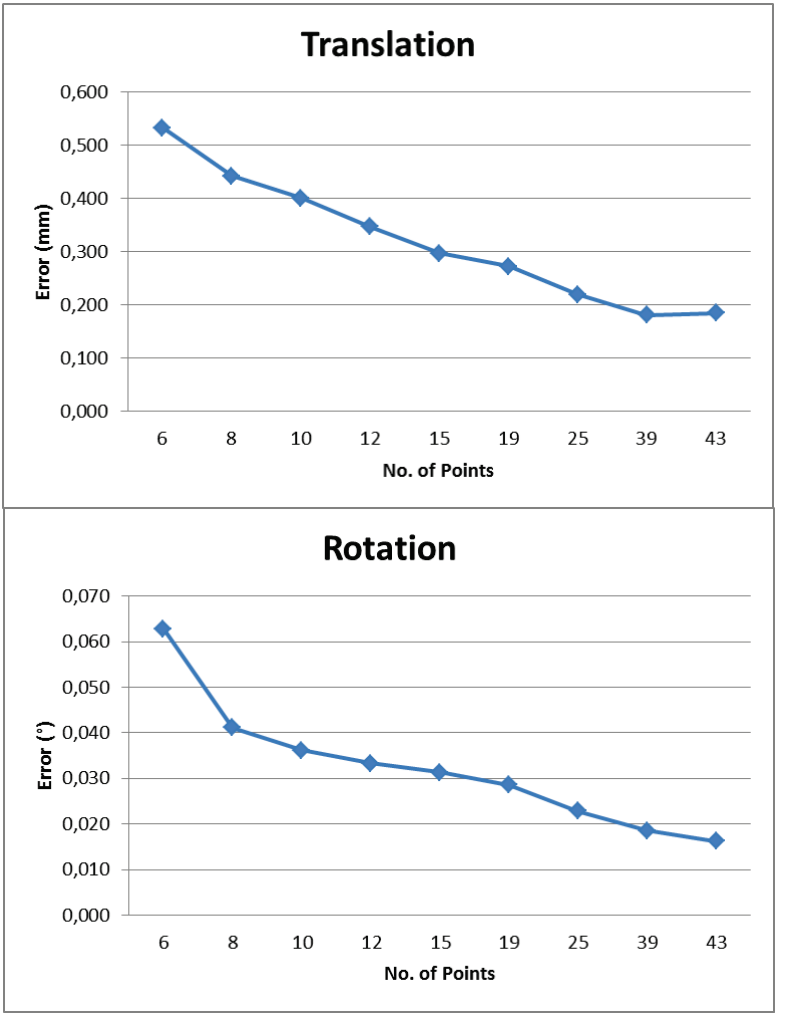

Figure 2.17: HPE errors with POSIT for different number of 2D-3D correspondences.

coplanar of all the facial points defined by the model, do not seem to make a significant improvement (the translation estimate even gets slightly worse).

Looking at these results, the 12-point model has been chosen as the reference model for the rest of this thesis. Working with more points would a priori give better accuracy in HPE, but the presented results have been obtained

in perfect tracking conditions. In real videos, 2D feature tracking algorithms are prone to some amount of error, and they usually perform best when the selected points are the most characteristic ones, which typically means choosing well-defined corners. The 12-point model consists of the inner and outer corners of the eyebrows and eyes, the corners of the mouth, and the corners defined by the eye nostrils and wings, as seen in Fig. 2.16e. Adding more points implies including points that, instead of being corners, are located on contours defined by the mouth, nose, eyes or eyebrows (see Fig. 2.16f-j). For tracking algorithms, these points usually perform slightly worse than specific corners because the point may drift along the contour during tracking, due to the appearance similarity of image patches along it. The 12-point model is also used in the rest of the experiments presented in this section.

2.3.2.2 Image Resolution

This study has been carried out on the synthetic head pose database, generating 2D ground truths for different image resolutions using integer pixel precision. All the selected resolutions correspond to a 16:9 aspect ratio, and they range from the 480×234 pixels of some touchscreen mobile phones to the 1920×1080 of Full HD. The aim of the study is to show the influence of the image resolution in POSIT HPE accuracy, assuming no subpixel accuracy is available.

The results of the experiment are provided in Fig. 2.18. The charts show a clear tendency: the higher the image resolution, the better accuracy in HPE given by POSIT. Nevertheless, the improvement is more significant in lower resolutions and the curve seems to flatten when the highest resolutions are approached, where the real 2D projections are better approximated by integer pixels. Translation errors range from 1.3mm of error for a 480×234 pixel resolution to 0.5mm of error for Full HD. Rotation errors on the other hand range from 0.5º of error for the smallest resolution to 0.12º for 1920×1080. For a 1280×720 video resolution, such as the one from the UPNA Head Pose Database presented in Section 2.1, translation and rotation average errors are approximately 0.6mm and 0.2°. This is the best HPE accuracy that could be expected from POSIT if 2D feature tracking is carried out using an algorithm with no subpixel accuracy.

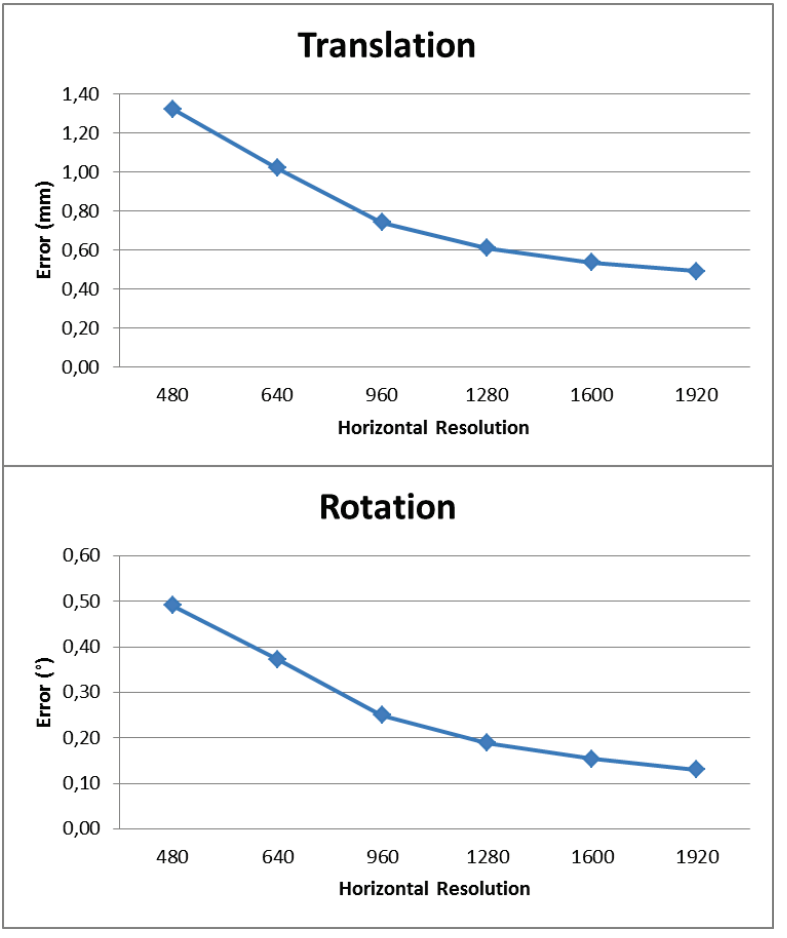

Figure 2.18: HPE errors with POSIT for different image resolutions with 16:9 aspect ratio.

2.3.2.3 Distance to Camera

The aim of this study is to analyze the effect of the distance from the head to the camera in the POSIT accuracy. For this purpose, specific image sequences have been created in the simulation environment, using always the mean shape of the BFM. Different translations in the Z axis, the optical axis, from the camera to the head have been set up, ranging from 20cm to 80cm at

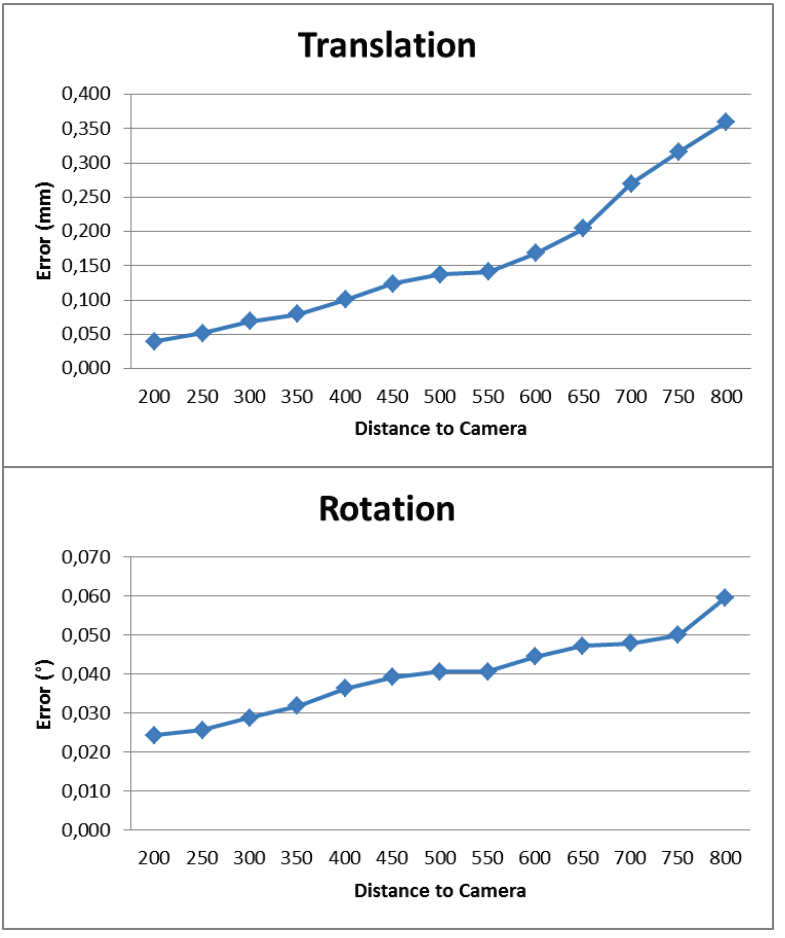

Figure 2.19: HPE errors with POSIT for different distances from the head to the camera.

5cm steps. For each distance, 1.125 head poses have been generated and projected onto the image, by varying the other two translations (X and Y) and the three rotations (roll, yaw and pitch). More specifically, a grid of 9 positions has been defined for each Z by setting X and Y translations at -10cm, 0cm, and 10cm from the optical axis. In each position, each of the angles has been set to five values: -30º, -15º, 0º, 15º and 30º. In summary, X and Y translations taking 3 different values each, and roll, yaw and pitch taking 5 different values each, adds up to 1.125 poses defined for each Z distance in the study. This amount and

variety of poses assures a good representation of POSIT performance against the distance between the head and the camera.

The results of the study are presented in Fig. 2.19. It is observed that, as the head is positioned further from the camera, the POSIT accuracy tends to decrease. The most likely explanation for this behavior is that, being the head always the same, the further it is from the camera the more coplanar the 12 points are seen with respect to the distance between the camera and the head. That is, the depth difference between the 3D model points becomes relatively smaller as the distance to the camera grows, therefore increasing the relative coplanarity between the points. This would make the first iteration of POSIT, the pose from orthography and scaling (POS), achieve a more accurate pose estimate as the head goes further from the camera, since the scaled orthographic projection would resemble more the "real" perspective projection. But precisely because of the iterative nature of POSIT, which consists in correcting these projections in each iteration in order to improve the pose estimate until a convergence is reached, the algorithm is able to achieve more accurate estimates when the head is closer to the camera. In these conditions, the relative coplanarity is reduced and the initial estimate given by POS can be further corrected in the iterative process, whereas having a higher relative coplanarity can make the POSIT converge faster, even in the first estimate given by POS, which does not assure a greater accuracy. This behavior is in accordance with the results obtained by DeMenthon & Davis in [16] for synthetic images of a cube and a tetrahedron.

The typical working distance when sitting in front of a computer is between 55cm and 60cm. Fig. 2.19 shows that, using the 12-point model described earlier, a top accuracy of around 0.15mm for translation and 0.04º for rotation could be expected at that distance, in perfect tracking conditions, if POSIT is used for HPE. This accuracy limit is considered reasonable for typical HPE applications, and therefore the POSIT algorithm is considered suitable for the purpose of this thesis. The next chapters research into different parts of the HPE process, but always rely on applying POSIT to the 2D-3D correspondences in order to get the pose estimate.

Bibliography of the Chapter

[1] M. Ariz, J. J. Bengoechea, A. Villanueva, and R. Cabeza, "A Novel 2D/3D Database with Automatic Face Annotation for Head Tracking and Pose Estimation," *Comput. Vis. Image Underst.*, no. Special Issue on Assistive Computer Vision and Robotics, 2016 (in press).

[2] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: anapproach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, 2000.

[3] "Flock of Birds," *Ascension Technology Corp., P.O. Box 527, Burlington, VT 05402.* .

[4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models-Their Training and Application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[6] I. Matthews and S. Baker, "Active Appearance Models Revisited," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 135–164, 2004.

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

[8] J. Bouguet, "Camera Calibration Toolbox for Matlab." [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[9] J. J. Bengoechea, R. Cabeza, and M. Ariz, "Comparativa de Algoritmos de Visión Monocular para la Estimación de la Posición de la Cabeza," Public University of Navarre, 2014.

[10] D. Roncal, R. Cabeza, and M. Ariz, "Técnicas de Seguimiento de Puntos Faciales y su Efecto en la Estimación de la Posición 3D de la Cabeza," Public University of Navarre, 2014.

[11] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D Face Model for Pose and Illumination Invariant Face Recognition," *2009*

Sixth IEEE Int. Conf. Adv. Video Signal Based Surveill., pp. 296–301, Sep. 2009.

[12] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid ICP algorithms for surface registration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[13] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," *ACM Trans. Graph.*, vol. 26, no. 3, p. 24, 2007.

[14] R. Segura, R. Cabeza, and M. Ariz, "Reconstrucción 3D de Modelos Personalizados de Cabeza para la Estimación de la Pose," Public University of Navarre, 2015.

[15] "http://faces.cs.unibas.ch/bfm/main.php." .

[16] D. F. DeMenthon and L. S. Davis, "Model-Based Object Pose in 25 Lines of Code," *Int. J. Comput. Vis.*, vol. 15, no. 1, pp. 123–141, 1995.

CHAPTER 3

3D Face Reconstruction

Due to the increasing and varied demands of computer vision and computer graphics applications, there has been a significant effort in developing reliable 3D object reconstruction methods in the last decade. Recovering the 3D structure of the face based on 2D images is especially important in many scenarios, since the face is currently one of the main channels of interaction. This chapter first reviews the most important publications in this field in Section 3.1. Different methods for recovering a sparse 3D shape (i.e. the 12-point model described in the previous chapter), suitable for HPE applications, are then proposed and described in detail in Section 3.2. All the methods are evaluated and compared among them on the same framework in Section 3.3 in order to find the most suitable one for HPE. Finally, a method for recovering a dense full-head 3D model from the sparse 3D shape is proposed and evaluated in Section 3.4. Conclusions for the whole chapter are summarized in Section 3.5.

3.1 Related Work

The face is an important channel of communication and this is the main reason why modeling 3D faces has been an important topic for the computer vision research community. It presents many applications, such as pose-invariant face recognition [1]–[3], age-invariant face recognition [4], facial expression analysis [5], virtual avatars [6], 3D person-specific game and movie character

generation [7], surgical simulation or model-based image coding [8], among others. It is thus of maximum importance in many computer vision face related applications to be able to build a generic 3D model valid in situations that are not part of the training. The model should be general enough to face illumination problems, different users, or facial expressions.

Possibly the most accurate way to obtain the 3D model of a specific face is to use special hardware or multi-camera systems, such as 3D laser scanners, structured light or stereo cameras [9]. The problem with this kind of approaches is the high cost of the devices and the complicated setup required in order to obtain the model. They usually require a tightly controlled environment, which limits their applicability. Moreover, there are cases where laser-scan based reconstruction cannot be performed, such as plastic surgery assistance. If a person's face has been damaged in an accident and requires plastic surgery, face reconstruction is often needed in order to assist the surgery. However, laser-scan based reconstruction is not a feasible approach, since the reconstruction has to be made from photos from before the accident. This is a clear case where 3D face reconstruction has to be performed using only information from 2D images.

There are different approaches for 3D face reconstruction from 2D images, such as shape-from-shading (SFS), parameterized appearance models (PAM), or structure-from-motion (SfM). SFS tries to recover the shape by exploring the shading information in the image, such as the intensity and its derivatives. It is usually based on using reflectance models, such as the Lambertian model or the specular reflectance model, to characterize the depth in the image by the intensity variations. The SFS estimates the direction of the illumination in the 2D image to generate a 3D surface. This approach was introduced by Horn [10] in 1970, and several attempts have been made to recover the shape from shading information in 2D images ever since [11]–[16]. Nevertheless, the skin has a very complex appearance to be modeled by shading, and these approaches have generally failed to achieve successful reconstruction results [12].

One of the most popular tools to model the shape and appearance of faces in images are PAMs, which are based on building a statistical model of shape and appearance from training sets of labeled data. PAMs encompass different techniques, such as eigentracking [17], [18], active shape models (ASMs) [19], [20], active appearance models (AAMs) [8], [21]–[25], active orientation models (AOMs) [26], or 3D morphable models (3DMMs) [2], [6], [27]–[29]. These methods are based on a previous learning stage using a training image set

in which key segmentation features have been annotated. ASM learns the statistical behavior of the face shape and the appearance of the neighborhood of each landmark, whereas AAM learns textures in the regions between landmarks in addition to the face shape, creating a combined model from the coupled relationship between shape and texture. PCA is usually applied to the training data in order to reduce its dimensionality and obtain the modes that best describe the observed variation. First studies with AAMs showed that the method was only efficient when working with frontal faces, because the appearance change due to 3D head pose was too large to be modeled properly [21], [23]. Later, Cootes et al. proposed a view-based AAM to deal with this issue [30], consisting of different appearance models for a set of discrete poses. The problem is that the amount of training images needed is largely increased and accuracy issues arise when the input pose differs from the set of poses used in the training. AOMs have been recently introduced and they follow the scope of AAMs, but instead of texture they use gradient orientations to build the model, and they incorporate an improved cost function, showing a better generalization to unseen identities.

While ASMs, AAMs and AOMs normally estimate 3D information just from 2D images, 3DMMs are usually built offline using directly 3D data obtained from 3D scanners or depth cameras in order to represent facial texture, shape, illumination and camera geometry with a large number of model parameters. Therefore, one of the main differences between ASM/AAM/AOMs and 3DMMs is that the formers' shape component is 2D whereas the latter's is 3D. 3DMMs are a promising approach because of their representational power, but they also show several drawbacks. First, as mentioned before, most of the times the model is built using expensive equipment (3D laser scans) that require complex setups. Second, 3D face databases are needed, and there are currently a limited number of them. Moreover, most of them show a lack of variability of facial expressions. And third, correspondences between models have to be found and all the training samples have to be registered between them, which is a complex procedure and very prone to errors.

When a new 2D image is given to a fitting algorithm using a 3DMM, it tries to find the parameters of the model that minimize the difference between the 2D image synthesized from the 3DMM and the 2D image received. The process is usually iterated until an optimal result is obtained. A complete 3D face can then be reconstructed applying the optimal parameters to the 3DMM. Nevertheless, the computational cost of the process is often very high due to the complexity of the model and the large number of parameters involved. Moreover,

the number of parameters to optimize being large often leads to local minima in the optimization process, which means that the resulting face is very close to the face used to initialize the process, typically the mean face of the 3DMM [31], [32]. Some works have focused on reducing the complexity of the morphable model in order to accelerate the fitting process. The original 3DMM is simplified by representing only the facial shape as described in [4], [33]–[35].

AAMs and 3DMMs have also been combined in some works in the literature, in an attempt to take advantage of the strengths of each and make up for the shortcomings. In [36], Faggian & Paplinski presented a fast method for fitting a 3DMM. AAMs were used to linearize the non-linear optimization problem of 3DMM fitting, which accelerated the typically slow process while preserving the representational power of the morphable model. Heo and Savvides [37] recently proposed a method to generate a 3DMM from 2D images, avoiding the need of high-cost equipment and complex setups to acquire the 3D data. They applied AAMs in order to extract facial shapes across different poses and then combined the 2D shapes to reconstruct a sparse 3D model that was then converted to a dense 3DMM using a subdivision algorithm. An average 3D depth-map was used to modify the resulting model in order to assure that a realistic facial structure was preserved. In [38], Xiao et al. presented what they called "Combined 2D+3D AAM" as an alternative to 3DMMs, trying to match their representational power while profiting from the fitting speed of the AAMs. The idea was to build a non-rigid 3D face model combining SfM and a fast 2D AAM facial feature tracker [24]. The 2D AAM was used to model the 3D phenomenon of faces moving across pose. The SfM algorithm was used to construct the corresponding 3D shape modes of the 2D AAM, and then these modes were used to constrain the AAM so that it would only generate model instances that could also be generated through the 3D modes, avoiding impossible shapes. The model construction was therefore performed just from 2D images, avoiding the more complex process of generating a 3DMM from range data.

SfM is a very common approach to reconstruct the 3D structure of a face when multiple frames of an image sequence are available and thus different views of the same head can be used, and it has been largely studied in the last fifteen years [39]–[62]. Typically, a certain set of 2D facial points is localized in all the frames so that their 3D shape can be recovered after applying SfM, which is usually called a factorization problem. The term SfM refers to the conceptual

approach defined in the previous lines, but it encompasses multiple algorithms that have been proposed in the past few years.

The first approach to SfM was made by Tomasi & Kanade in the early 90's in [39], where they proposed a robust factorization algorithm to recover a shape matrix and a motion matrix from an orthographic camera, using singular value decomposition (SVD). This work has been a starting point for other studies that have developed the technique to deal with non-rigid structures [42], [43], [50], [51], [55] or missing data [40], [45], [59], [61].

In one of the first works that considered the 3D object as non-rigid [42], the observed shapes were modeled as linear combinations of a few basic shapes that defined the principal deformation modes. Working under a weak perspective camera model, the method factorized the 3D shape and motion of the object using the rank constraints in the camera rotation matrices. Later, Torresani et al. [55] assumed a Gaussian prior for the shape coefficients and an expectation-maximization (EM) algorithm was applied to solve the optimization. Recently, Akhter et al. [60] proposed a dual approach that represented the 3D structure as an evolving structure in the trajectory space, which allowed defining an object-independent basis. Instead, the evolving 3D structure was represented as a linear combination of basic trajectories. Moreover, instead of applying PCA to build the object independent basis, they proposed using the discrete cosine transform (DCT) in order to reduce the number of unknowns and increase the stability of the estimation.

Among possible factorization algorithms to recover the shape and motion of 3D structures in SfM, EM [45], [46], [55], alternated least squares (ALS) [40], [44], [52] and nonlinear least squares (NLS) are possibly the ones that have proven to be the most effective. One of the main focuses in these works has been to deal with noisy data, which means working with tracking inconsistencies or missing data. In these situations, SVD is known to be inefficient and not able to achieve good reconstruction, and the cited papers have tried to incorporate other techniques with the goal of extending SfM to cases with missing data or inaccurate 2D tracking. The work by Torresani et al. [44] was one of the first to address this issue, and they presented very promising results using an alternation approach. Koo & Lam [57] proposed a reconstruction method based on the similarity-transform that dealt with any pose variation in the face images at a high computational cost. To reduce this cost, later Sun et al. [62] introduced the NLS model in a similar optimization framework to estimate the depth values of the facial feature points. Lee et al. [61] presented a method that used a shape

conversion matrix (SCM) to estimate the true location of the 2D points in the presence of self-occlusions, as an alternative to be incorporated in general SfM methods. Buchanan & Fitzgibbon [47], [49] showed that second-order optimization converged in a more reliable way than ALS, but realized that minimizing the reprojection error was not enough to achieve reliable reconstructions. There emerged the idea of the need of establishing constraints to the object deformation based on the prior knowledge of the 3D structure, and finding ways to smartly use this information.

Several works have focused on the constraints applied in the recovery of the shape, showing that the rank constraints originally proposed by Bregler [42] were not sufficient in order to accurately reconstruct the object. Brand [43] introduced some extra constraints that forced the deformation of the object relative to the mean shape, the starting point in the optimization, to be small. Later, Xiao et al. [51] worked on the constraints applied to the shape basis, proposing some extra ones that allowed them to achieve better reconstruction results. The basic assumption under this proposal was that there were as many image frames as basis shapes, where these shapes were known to be independent, and this number was estimated along the process. Nevertheless, later on Brand [50] showed that this algorithm is not valid in the presence of noisy data or when the number of basis shapes is not correctly estimated.

Although the idea of finding suitable constraints that assure a meaningful reconstruction of the 3D shape seems promising, there are scenarios where this deformation is inherently unconstrained. This is often the case when working with 2D features from monocular image frames. In these scenarios, several approaches have been made in order to avoid degenerate shape configurations using prior knowledge of the 3D structure, such as the one by Torresani et al. mentioned before [46], [55], where they applied EM and assumed Gaussian priors over the deformation parameters. In [48], Del Bue et al. proposed a factorization method that focused on some priors over the rigid component of the object, specifically applying them on some of the shape points that were considered more rigid than the rest. Olsen & Bartoli [53] developed a factorization method that assumed temporal smoothness and continuous variation in the reconstruction of the 3D structure. In [56], Del Bue proposed incorporating prior knowledge of the feasible 3D shape configurations and using this information as a regularization term for the rigid component of the deformable object. He introduced this prior into an intermediate affine solution instead of the final reconstruction. More recently, Gonzalez-Mora et al. [59]

proposed an incremental SfM approach that incorporated a similar prior on the feasible 3D shape configurations into the final Euclidean linear basis reconstruction, which provided the method with more robustness to noise, missing data and degenerate solutions.

One of the main differences between the three general techniques mentioned so far (i.e. SFS, 3DMMs and SfM) is that SFS and 3DMMs generate dense 3D point clouds, whereas SfM generally produces a sparse 3D representation, that is, a reduced set of 3D facial features representing the face. This characteristic makes SfM a better approach for real-time applications, such as HPE, because much less information is handled. However, some applications such as face recognition, usually achieve better results using dense 3D representations, since having information about a larger amount of facial points becomes critical.

Generic elastic models (GEMs) are another interesting approach to reconstruct 3D shapes from 2D observations that has recently been introduced

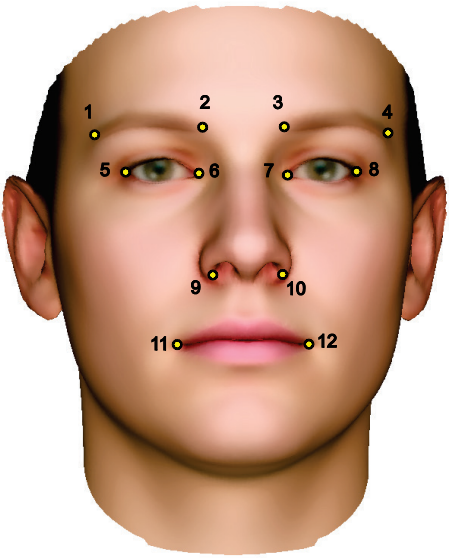

Figure 3.1: The 12-point model defined for the 3D face reconstruction methods proposed in this thesis.

and seems very promising [63], [64]. The underlying assumption in GEMs is that depth information in a face (the *Z* component) is not significantly discriminative among different individuals, at least from the same ethnic group [63]. This means that, if the faces of different individuals are correctly aligned, the depths of any correspondent facial points are not very different among faces. Therefore, the depths of a set of facial points could be approximated by another person's depths if they are known, or by the depths of the same anatomical points obtained from a generic 3D face model. The GEM is thus adjusted by elastically deforming the chosen generic depth map according to the (*X,Y*) positions of the points in an input frontal image of the face. The main advantages of GEMs are that they can generate the 3D shape from a single input image and that they entail very low computational expense. It is still unclear whether the GEM's assumption about depths and all its implications are a valid approach in any scenario. For instance, it is derived from it that depth information in faces is not significantly discriminative for modeling 2D pose variability.

This thesis is mainly devoted to contribute to HPE methods, in which there are often real-time requirements and the 3D face reconstruction is usually carried out from video sequences containing different views of the head. That is why SfM or GEMs result of more interest than SFS or 3DMMs for our purpose, and will thus receive more attention in the following section.

3.2 Proposed Reconstruction Methods

This section aims to describe the different face reconstruction methods that are proposed in this thesis. All of the approaches recover a 3D shape only based on 2D observations. Some of them only require a frontal view of the face, whereas others make use of different poses in the search for more accurate reconstructions. All of them are presented based on the 12-point model defined in the previous chapter and shown in Fig. 3.1, and they will be evaluated and compared between them in Section 3.3. For the moment, an exhaustive description of each of the methods is presented in the following lines.

3.2.1 Cylindrical Head Model

The first reconstruction method proposed is based on using a cylindrical head model (CHM). This has been a classical approach in the literature for HPE methods [65]–[68]. The surface of the face is usually approximated by an elliptic

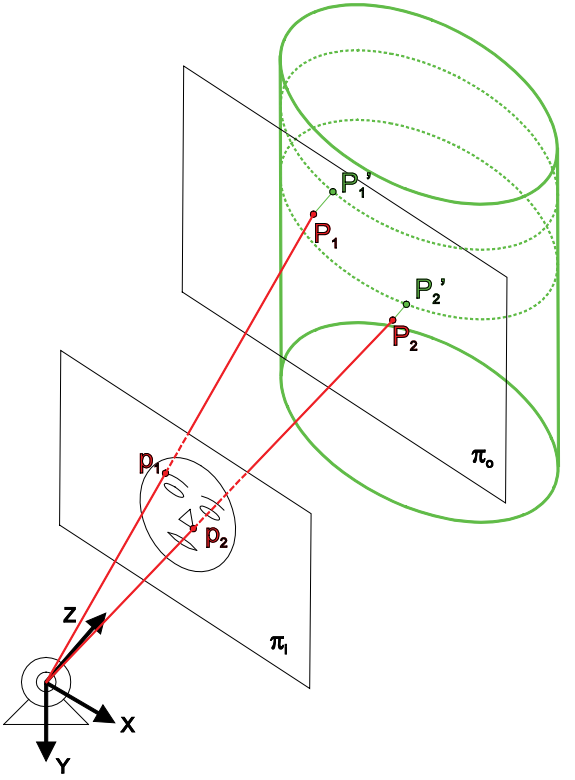

Figure 3.2: Scheme of 3D face reconstruction from 2D images using the CHM. Two example image facial feature points are back-projected to the object plane, and final corresponding 3D points in the CHM are obtained by orthographic projection.

cylinder. Once the coordinates of the facial points on the *XY* plane are determined, the depths in the *Z* axis are given by the corresponding points in the cylindrical surface. We propose obtaining the *X* and *Y* coordinates of the 3D shape by back-projecting the 2D observations from a frontal view of the face, as shown in Fig. 3.2. The image plane (π_i) and object plane (π_o) are represented parallel to each other. Two example image facial feature points (p_1 and p_2) are back-projected to the object plane generating P_1 and P_2, 3D points at the same depth. The specific depth of each point is then obtained by orthographically projecting them onto the cylinder surface, generating the 3D corresponding points in the CHM (P_1' and P_2').

In this configuration, the camera and object coordinate systems are aligned and image and model coordinates in the *XY* plane can be related by simple scaling factors if we assume there is no distortion. In those conditions, image and object coordinates are expressed by the following equations defining a basic pinhole camera model

$$x = \frac{fc}{Z_0} X \quad , \quad y = \frac{fc}{Z_0} Y \qquad (3.1)$$

where fc is the focal length of the camera and Z_0 the distance from the camera to the object. The term $\frac{fc}{Z_0}$ is thus a scaling factor expressed in $\frac{pixel}{mm}$ that relates the image and object coordinate systems. In order to know the exact distance from the face to the camera for a given frontal view, we would need to know the exact dimensions of the face, which is not possible unless a 3D scan of the user's head is available. We propose an alternative that consists in assuming certain global fixed dimensions for the head, obtained from an average head calculated from the 3D scans of 200 users, i.e. the mean shape of the BFM already mentioned in this thesis [69]. This is equivalent to setting the distance Z_0 at which the object plane (π_o) is located in Fig. 3.2, which in turn determines the cutting points for the back-projection lines (P_1 and P_2 in that figure). The fixed dimensions that will give us the scale are the distance between the outer corners of the eyes for a horizontal scale factor, and the distance from one of the outer eye corners to one of the mouth corners for a vertical scale factor. Let's denote $\boldsymbol{x}_f = (x_f, y_f)$ and $\boldsymbol{X}_f = (X_f, Y_f, Z_f)$ the image and model coordinates of the 2D and 3D corresponding facial feature points defined in Fig. 3.1, with $f = 1 \dots 12$. Both external eye corners ($\boldsymbol{x}_5$ and $\boldsymbol{x}_8$ in the image, and $\boldsymbol{X}_5$ and $\boldsymbol{X}_8$ in the model) and one of the mouth corners ($\boldsymbol{x}_{11}$ in the image and $\boldsymbol{X}_{11}$ in the model) are used as reference points to find the horizontal and vertical scaling factors (s_h, s_v). Having manually annotated $\boldsymbol{X}_5$, $\boldsymbol{X}_8$ and $\boldsymbol{X}_{11}$ in the mean shape of the BFM, the scaling factors for a given frontal view of the head can be calculated as

$$s_h = \frac{|X_5 - X_8|}{|x_5 - x_8|} \quad , \quad s_v = \frac{|Y_5 - Y_{11}|}{|y_5 - y_{11}|} \qquad (3.2)$$

The twelve 2D image points are then translated so that their origin is set at the right eye's external corner, by subtracting $\boldsymbol{x}_5$ to every one of them. The *X* and *Y* coordinates in the object reference system for any facial point *f* are then found as

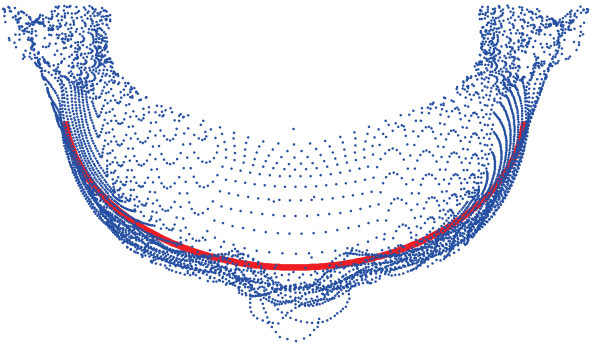

Figure 3.3: Top-view of the cylinder adjusted to the 53.490 vertices of the BFM by minimizing their distance to the surface of the elliptic cylinder.

$$(X_f, Y_f) = (X_5 + s_h x_f,\ Y_5 + s_v y_f) \tag{3.3}$$

The depth for each point is then found by making the point lie on the cylinder surface. The dimensions of the cylinder have been previously defined by adjusting a general cylinder, with its axis aligned with the vertical Y axis, to the BFM [69], a 3DMM described in the previous chapter. This is done by finding the cylinder that minimizes the distance from each of the 53.490 vertices of the mean shape of the BFM to the surface of the cylinder, as shown in Fig. 3.3. The depth of the reconstructed 3D shape points is then easily found using the following equation

$$Z_f = \sqrt{r^2 - {X_f}^2} \tag{3.4}$$

where r is the radius of the adjusted cylinder. Note that Y_f does not appear in the equation, since the cylinder is aligned with the Y axis.

The process just described for obtaining the cylinder implies using a 3D morphable face model (the BFM) because we have chosen to do so in order to set the cylinder dimensions. However, it should be noted that any cylinder could be defined, with reasonable dimensions so that it approximates a human head,

and this 3D face reconstruction method would not require having any 3DMM stored in advance.

3.2.2 Generic 3D Morphable Face Model

The second approach presented consists in using a generic 3D morphable face model to approximate any input face, specifically the mean shape of the BFM. The resultant 3D shape is obtained simply by annotating the 12 facial points of interest in the mean shape of the BFM (the reader can refer to Fig. 3.1 to see it). The main difference of this approach with respect to the CHM is that the depths of the 3D points follow a 'human' pattern instead of a cylindrical surface. However, in this case the X and Y components of the 3D shape are not obtained based on the back-projections of 2D observations, but are fixed generic coordinates applied to any user. In other words, this approach does not consist in recovering a 3D model from image observations, but in using the same 3D shape belonging to a generic 3DMM for modeling any user's face.

3.2.3 Scaling Approach

The scaling approach is based on applying scaling factors to the 12-point 3D shape of the generic 3DMM presented above. The scales will be calculated based on 2D observations. First, a simpler approach based on calculating the aspect ratio between X and Y spatial components is described. Then, a more complex approach based on obtaining independent scale factors for the three spatial components, including Z, is presented.

3.2.3.1 Aspect Ratio

The aspect ratio between X and Y spatial components can be obtained from a single frontal view of the face. Using the same three facial feature points as reference as for the CHM approach (i.e. outer corners of the eyes and one corners of the mouth), annotated both in 2D in the image and in 3D in the BFM, a vertical and a horizontal scaling factor (s_v and s_h) can be calculated for a given frontal view of the head using equation (3.2). In order to keep the same global dimensions as the BFM, thus making just an aspect ratio adjustment, the scaling factors are normalized as

$$s_h{}' = \frac{s_h}{mean(s_h, s_v)} \quad , \quad s_v{}' = \frac{s_v}{mean(s_h, s_v)} \qquad (3.5)$$

The X and Y components of the mean shape of the BFM are then scaled as

$$X_f^{sc} = X_f \cdot s_h' \quad , \quad Y_f^{sc} = Y_f \cdot s_v' \tag{3.6}$$

where X_f^{sc} and Y_f^{sc} are the new scaled X and Y spatial components of a given feature point (X_f, Y_f). Note that the Z_f components of the 3D shape remain unaltered and are those of the generic shape of the BFM.

3.2.3.2 Full Scaling

This approach consists in finding an optimal independent scale factor for the three spatial components (i.e. X, Y, Z) of the 3D shape of the generic BFM. Just a frontal view of the face was necessary in the previous scaling approach to find the aspect ratio for the XY plane, since this plane is parallel to the image plane in those conditions. Nevertheless, multiple views of the head in different poses are necessary in order to find an optimal scale for the out-of-plane Z component.

Taking this into account, a multiple-view optimization framework is proposed in Algorithm 1, in which the 3D shape points $\boldsymbol{X}_f$ are iteratively adjusted until an optimal 3D model $\widehat{\boldsymbol{X}}_f$ is reached. The 3D shape is initialized using the annotated mean shape of the BFM. At each iteration of the optimization process, the POSIT algorithm is first applied to recover the pose using the current 3D shape points $\boldsymbol{X}_f^i$ and their correspondent 2D observations in the image $\boldsymbol{x}_f$ for all the views of the face. Then, using the estimated pose, the 3D shape is projected onto each training image and the 2D projections $\boldsymbol{x}_f'$ are compared with the real 2D corresponding points in terms of average Euclidean distance

$$e = EuclDist(\boldsymbol{x_f}, \boldsymbol{x_f}') = \frac{1}{F}\sum_{f=1}^{F}\sqrt{\left(x_f - x_f'\right)^2 + \left(y_f - y_f'\right)^2} \tag{3.7}$$

where F is 12, the total number of facial feature points. The metric e is thus an image error metric that represents the convergence of the method. The 3D shape is iteratively adjusted until the distance between the 2D projections and the real 2D observations is minimized across all the views in the training set.

Algorithm 1 Multi-view based 3D model fitting optimization framework

1: **Input:** $\{\boldsymbol{X}_f: (X_f, Y_f, Z_f)\}_{f=1}^{12}, \{\boldsymbol{x}_f: (x_f, y_f)\}_{f=1}^{12}$
2: Initialize image error $e = \infty$
3: **repeat until** $e < \varepsilon$
4: Modify $\boldsymbol{X}_f^i$, being i the current iteration
5: Estimate the head pose using POSIT with $\boldsymbol{x}_f$ and $\boldsymbol{X}_f^i$
6: Calculate $\boldsymbol{x}_f{}'$ by projecting $\boldsymbol{X}_f^i$ in the estimated pose
7: Image error: $e = EuclDist(\boldsymbol{x}_f, \boldsymbol{x}_f{}')$
8: **end repeat until**
9: **Output:** $\{\hat{\boldsymbol{X}}_f: (\hat{X}_f, \hat{Y}_f, \hat{Z}_f)\}_{f=1}^{12}$

This optimization scheme will be used for other multiple-view based 3D face reconstruction approaches proposed in this thesis, as it will be explained later. The 3D shape will be modified according to the deformation proposed in each approach, in an unconstrained nonlinear optimization framework following the Nelder-Mead simplex method [70]. In this case, an independent scaling factor for each spatial component is calculated at each iteration of the optimization framework, and the resultant 3D model points are calculated as

$$X_f^{sc} = X_f \cdot s_x \quad , \quad Y_f^{sc} = Y_f \cdot s_y \quad , \quad Z_f^{sc} = Z_f \cdot s_z \qquad (3.8)$$

where s_x, s_y and s_z are the scaling factors for the X, Y and Z spatial dimensions respectively, and X_f, Y_f and Z_f are the spatial components of each annotated 3D feature point f of the mean shape of the BFM.

3.2.4 PCA-Based Model Fitting

The PCA-based approach uses a parametric face model defined on a PCA basis in order to adjust the parameters so that an error metric is minimized based on the 2D observations. In order to achieve a better fitting, the multi-view optimization framework described in Algorithm 1 will be adopted. First, the parametric face model is explained, and then a regularization term to constrain the model deformation and avoid degenerate solutions is presented. Preliminary experimental results showing the performance of both approaches with respect to

the number of modes of the PCA used in the optimization are presented and discussed.

3.2.4.1 Parametric Face Model

The BFM, i.e. the 3DMM presented by Paysan et al. in 2009 [69], is a parametric face model built based on training data obtained from the 3D scans of 200 subjects, 100 females and 100 males between 8 and 62 years old, most of them Caucasian. All the scans contained a neutral facial expression and were registered to ensure an optimized anatomical point correspondence between faces. The faces were parameterized as triangular meshes after registration, resulting in m =53,490 vertices described by a shape vector $(x_i, y_i, z_i)^T \in \mathbb{R}^3$ with an associated texture vector $(r_i, g_i, b_i)^T \in [0,1]^3$, where $i = 1 \dots m$. PCA was then applied to the data in order to create an orthonormal basis of n =199 principal components of texture and shape that make up a parametric face model $\mathcal{M}_{\{s,t\}}$ that consist of

$$\mathcal{M}_s = (\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s, \boldsymbol{U}_s) \quad , \quad \mathcal{M}_t = (\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t, \boldsymbol{U}_t) \tag{3.9}$$

where $\boldsymbol{\mu}_{\{s,t\}} \in \mathbb{R}^{3m}$ are the mean, $\boldsymbol{\sigma}_{\{s,t\}} \in \mathbb{R}^{n-1}$ are the standard deviations, and $\boldsymbol{U}_{\{s,t\}} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_n] \in \mathbb{R}^{3m \times n-1}$ are the orthonormal basis of principle components of shape ($\boldsymbol{s}$) and texture ($\boldsymbol{t}$). Any face in the PCA basis can then be represented as a linear combination of the principal components

$$\begin{aligned} \boldsymbol{s}(\boldsymbol{\alpha}) &= \boldsymbol{\mu}_s + \boldsymbol{U}_s \, diag(\boldsymbol{\sigma}_s)\boldsymbol{\alpha} \\ \boldsymbol{t}(\boldsymbol{\beta}) &= \boldsymbol{\mu}_t + \boldsymbol{U}_t \, diag(\boldsymbol{\sigma}_t)\boldsymbol{\beta} \end{aligned} \tag{3.10}$$

where $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]$ is a vector of n coefficients that define the shape and $\boldsymbol{\beta} = [\beta_1, \dots, \beta_n]$ a vector of n coefficients that define the texture. By varying those coefficients, all the possible faces in the PCA subspace can be obtained.

The PCA-based model fitting to recover a 3D shape of the face is thus carried out by introducing $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_L]$ coefficients in the optimization process described in Algorithm 1, being $L \in [1, n]$ the number of principal components used in the deformation. PCA analysis is useful because it sorts the principal components based on their associated eigenvalues, and thus based on the amount of variance they model. Therefore the reconstruction can be made with a lower number of modes without a significant loss of information. The

shape $\boldsymbol{s(\alpha)}$ in equation (3.10) contains the 53,490 vertices that define the complete head model. Since there is an exact anatomical correspondence between the vertices from any face generated in the PCA subspace, the twelve facial feature points of interest $\boldsymbol{X_f}$ can be annotated as vector indexes and easily recovered for any set of α coefficients. The optimal coefficients $\hat{\boldsymbol{\alpha}}$ that achieve the final 3D shape $\hat{\boldsymbol{X}}_f$ are then found following the scheme laid out in Algorithm 1, and the final shape is recovered as

$$\hat{\boldsymbol{X}}_f = \boldsymbol{\mu_X} + \boldsymbol{U_X}\, diag(\boldsymbol{\sigma_X})\hat{\boldsymbol{\alpha}} \qquad (3.11)$$

3.2.4.2 Regularization Term

The problem with 3D face reconstruction methods that are based on deforming the model on a PCA basis is that they can produce degenerate solutions (i.e. shape configurations considered impossible for real faces) if the deformation is totally unconstrained. Degenerate solutions are usually linked to high α values, since the resultant model differs more from the average configuration, for which all α's are 0. Therefore, we propose introducing a regularization term based on the shape coefficients in the optimization framework in order to constrain the model deformation and assure feasible shape configurations. This term is introduced in the calculation of the Euclidean error metric that will be minimized during the optimization

$$e = \frac{1}{F}\sum_{f=1}^{F}\sqrt{\left(x_f - x_f{}'\right)^2 + \left(y_f - y_f{}'\right)^2} + a\frac{1}{L}\sum_{l=1}^{L}{\alpha_l}^2 \qquad (3.12)$$

where $a\frac{1}{L}\sum_{l=1}^{L}{\alpha_l}^2$ is the regularization term, being a a constant that controls the level of regularization introduced. A high value of a will result in a recovered 3D model close to the mean shape of the BFM, whereas a small a will approximate the unconstrained fitting described in Section 3.2.4.1. Note that the sum of squares of the coefficients in the regularization term is averaged so that the number of principal components used in the deformation does not affect the error. The specific optimization scheme is detailed in Algorithm 2.

3.2.4.3 Preliminary Experimental Results

The PCA-based model fitting approach has been presented with the alternative of introducing a regularization term in the optimization process.

Algorithm 2 Multi-view PCA-based 'regularized' model fitting optimization framework

1: **Input:** $\boldsymbol{\alpha} = 0, \{\boldsymbol{x}_f : (x_f, y_f)\}_{f=1}^{12}$
2: Initialize image error $e = \infty$
3: **repeat until** $e < \varepsilon$
4: Modify $\boldsymbol{\alpha}^i$, being i the current iteration
5: Obtain $\boldsymbol{X}_f^i = \boldsymbol{\mu_X} + \boldsymbol{U_X}\, diag(\boldsymbol{\sigma_X})\boldsymbol{\alpha}^i$
6: Estimate the head pose using POSIT with $\boldsymbol{x}_f$ and $\boldsymbol{X}_f^i$
7: Calculate $\boldsymbol{x}_f{}'$ by projecting $\boldsymbol{X}_f^i$ in the estimated pose
8: Image error: $e = EuclDist(\boldsymbol{x}_f, \boldsymbol{x}_f{}') + a\frac{1}{L}\sum_{l=1}^{L}{\alpha_l}^2$
9: **end repeat until**
10: Obtain the optimal shape $\widehat{\boldsymbol{X}}_f = \boldsymbol{\mu_X} + \boldsymbol{U_X}\, diag(\boldsymbol{\sigma_X})\widehat{\boldsymbol{\alpha}}$
10: **Output:** $\{\widehat{\boldsymbol{X}}_f : (\hat{X}_f, \hat{Y}_f, \hat{Z}_f)\}_{f=1}^{12}$

Besides, the orthonormal basis of the BFM consists of 199 principal components, from which a certain number $L \in [1,199]$ will be used for the reconstruction. Obviously, the more modes introduced in the reconstruction, the longer the optimization of the coefficients will be, since each coefficient is a parameter to optimize. In order to analyze the influence in the reconstruction accuracy of both the number of modes optimized and the use or not of regularization, a preliminary study has been carried out and is presented here. The objective of the study is to select the best configuration for the PCA-based fitting approach, which will be the one used for the performance comparison carried out in Section 3.3. These preliminary results have been obtained on the synthetic head pose database presented in Chapter 2, using 90 random frames per user for face reconstruction and the complete database for evaluation. The specific framework is the same as the one employed for the evaluation of the rest of the methods and, instead of giving details here, it will be thoroughly described in Section 3.3.1.

Results are presented in Fig. 3.4. Average rotation and translation HPE errors have been obtained on the synthetic database using POSIT with the 2D ground truth and the reconstructed 3D shape. Model fitting errors have been obtained as the point-to-point average Euclidean distance between the 3D shapes of the reconstructed model and the real head. These errors have been calculated for the raw reconstructed shape, and for the reconstructed shape optimally registered to the real model using Procrustes analysis, which gives optimal

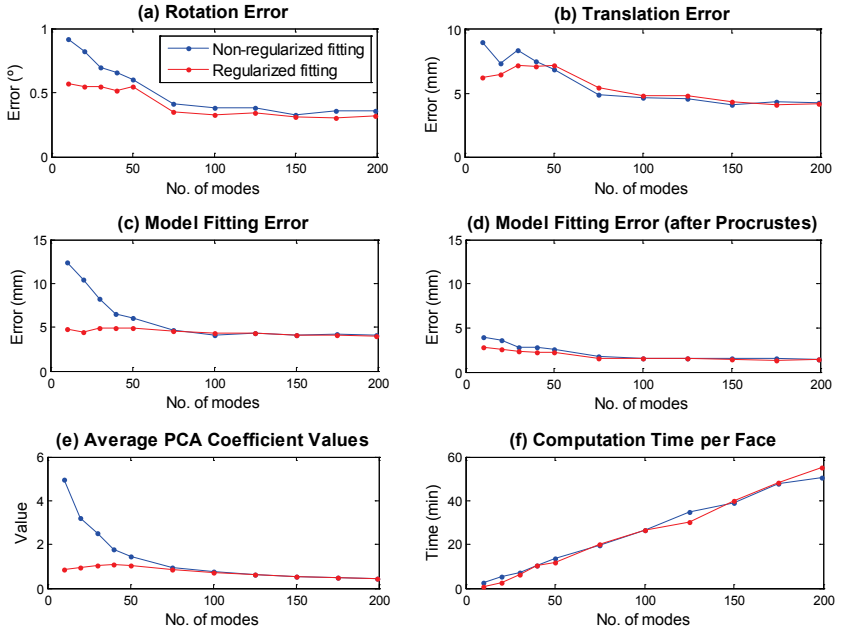

Figure 3.4: Results for the preliminary study of the PCA-based model fitting, analyzing the performance of the method using regularization (red curve) and without using it (blue curve) against the number of principal components used in the face reconstruction. (a) Rotation average error; (b) Translation average error; (c) Point to point 3D model error; (d) Point to point 3D model error after registration of the reconstructed shape with the real 3D shape; (e) Average value of the $\boldsymbol{\alpha}$ coefficients that define the shape; (f) Average computation time per reconstructed face.

alignment between shapes by adjusting rotation, translation and scale. Besides, the average value of the optimal $\boldsymbol{\alpha}$ coefficients that represent the shape has been calculated. Finally, the computation time employed in the face reconstruction has been estimated.

3.2.4.4 Discussion

Two clear tendencies are observed in the results presented in Fig. 3.4. On the one hand, the more principal components used for the reconstruction, the more accurate results are obtained. On the other hand, results are generally better when the regularization term is introduced in the optimization process, especially when few principal components are used for the deformation.

When the regularization term is introduced in the optimization, the error curves are quite stable regardless the number of principal components used. However, when no regularization is used in the optimization, all the error curves show a significant drop in the 10-50 number of modes range, and stabilize when 75 or more modes are used, showing a performance similar to the regularized fitting for this last range. The average value of the coefficients that multiply the principal components in the PCA basis confirms this behavior, since it remains stable slightly under 1 for the whole range when regularization is employed, whereas it goes up to five times that value when only ten modes and no regularization are applied. This is a clear indicator of a degenerate solution in the 3D face reconstruction because it has to be noted that the synthetic faces have been generated with random coefficients normally distributed with zero mean and unitary variance, as explained in detail in Chapter 2. Model fitting error is significantly reduced when the reconstructed faces are registered to the real synthetic ones, and the difference between regularized and non-regularized fitting is smaller. The same behavior in the mentioned ranges is still observed, but in a smaller scale. However, it is likely that registration is partially correcting model fitting error, especially when this one is bigger, as in the case of using few modes and no regularization. Finally, it can be observed that the computation time employed in the face reconstruction increases linearly with the number of modes optimized, and the regularization term does not seem to affect this time.

It is important to note that in this study face reconstruction has been carried out in perfect tracking conditions (i.e. using the 2D ground truth of the database). It is likely that the regularization term becomes more important in noisy tracking conditions, since inaccurate 2D observations may easier lead to degenerate face reconstructions.

Considering all the above, we can draw two main conclusions: 1) it is beneficial to introduce a regularization term that constrains the deformation in the PCA-based model fitting approach; 2) both the 3D fitting error and the HPE error are reduced as more principal components are optimized, at the cost of linearly incrementing the face reconstruction time. Therefore, a compromise has to be found, and the choice should be made based on the requirements of the target application. In our case, we will choose the regularized fitting using 75 modes for the performance comparison with the rest of the methods that will be carried out in Section 3.3.

3.2.5 Back-Projection-Based Fitting

This model fitting approach is based on the back projection framework described for the CHM in Section 3.2.1. The *X* and *Y* components of the 3D shape points are similarly obtained by back-projection from a frontal view of the face using equations (3.2) and (3.3). However, instead of approximating the depth components to a cylindrical surface as in the CHM, three different approaches using the BFM for the calculation of the *Z* spatial components are proposed in the following subsections.

3.2.5.1 Generic Depths

The most basic approach for obtaining depth components for the 3D shape points consists in assigning a generic depth map obtained from the mean shape of the BFM. The same anatomical twelve 3D facial points have previously been annotated in the generic morphable model, and the depths of these points are used for the 3D face reconstruction of any new user. This approach is basically a GEM [63], [64], as described in Section 3.1, where the face reconstruction is achieved by elastically deforming the chosen generic depth map according to the 2D (*x*,*y*) positions of the points in an input frontal image of the face.

3.2.5.2 Depth Interpolation

Another approach consists in assigning the depths based on the intersection points between the back-projections of the 2D image observations and the 3D generic model (the BFM). This is an interesting approach when the anatomical corresponding points of the 2D observations are unknown; using just a frontal image, the image points are back-projected to a generic 3D model and the depths for the corresponding 3D shape points are obtained as the depths at which the back-projection lines intersect with the 3D surface of the model.

As it has been said, *X* and *Y* coordinates are obtained using equations (3.2) and (3.3). Let us denote $\{\boldsymbol{X}_i\}_{i=1}^{m}$ the whole 3D mesh of the generic 3D model, with m=53,490 in the case of the BFM. The previously obtained *X* and *Y* spatial coordinates do not necessarily match those of any model point from the 3D mesh, so *Z* must be interpolated. The *K* 3D points $\{\boldsymbol{X}_k\}_{k=1}^{K}$ with the smallest Euclidean distance to the back-projection line are obtained from the model, and used to calculate each depth as

$$Z_f = \frac{\sum_{k=1}^{K} \frac{1}{d_k} Z_k}{\sum_{k=1}^{K} \frac{1}{d_k}} \tag{3.13}$$

where d_k is the Euclidean distance from $\boldsymbol{X}_k$ to the (X_f, Y_f) back-projection 3D line for each feature *f*. *K* has been set to 9 in our method based on preliminary experiments, where it has also been observed that the value of this parameter has a small impact and is not critical for the performance of the method.

3.2.5.3 Depth from Different Views

The last approach proposed to estimate the depths of the points in back-projection based fitting consists in using the multi-view based optimization framework from Algorithm 1. The *X* and *Y* spatial components have previously been obtained by back-projection using a frontal view of the face. However, an accurate estimation of point depths is hard to obtain from frontal views, where the *Z* axis of the model is aligned with the optical axis of the camera and therefore depth information is lost. Using different views of the face produced by out-of-plane rotations, we can assume that a more accurate estimation of face depths can be achieved. The optimization scheme from Algorithm 1 is in this case applied by iteratively modifying the *Z* components of the twelve facial points while *X* and *Y* coordinates are fixed (previously obtained by back-projection), until optimal depths that minimize the reprojection error are found. The initial depth values for the optimization loop are the generic depths from Section 3.2.5.1, obtained from the mean shape of the BFM.

3.2.6 Bundle Adjustment

Bundle adjustment (BA) is a popular technique usually employed as the final step of many 3D object projective reconstruction algorithms in computer vision [71]. It uses an image sequence that depicts a set of 3D points of the object from different viewpoints in order to refine in an optimal way both the 3D structure that describes the scene geometry and the extrinsic and intrinsic camera parameters that describe the relative motion and optical characteristics respectively. The name of the method describes the process of adjusting the bundle of rays between the 3D points in the object and the optical center of the camera in the different positions. The optimization is carried out by minimizing the reprojection error between the observed and predicted image points using

non-linear least squares algorithms, from which Levenberg-Marquardt (LM) is considered to be the most successful one.

BA is usually employed as a final step of any reconstruction algorithm because it requires a good initialization in order to provide accurate results. Moreover, it can be a large minimization problem due to the amount of parameters to optimize. Nevertheless, it is a widely used algorithm because it can effectively handle missing data and always provides a true maximum likelihood estimate. Among the different implementations that have been proposed for BA, possibly the most standard method for optimizing SfM problems in computer vision is the sparse bundle adjustment (SBA). The sparse implementation makes special emphasis on flexibility and performance efficiency by applying a specific variant of the LM algorithm that takes advantage of the sparse block structure of the problem, which basically consists in the lack of interaction among parameters for different 3D points and cameras. This results in a significant reduction of the computational cost of the reconstruction.

We have chosen to implement the SBA [72] approach for the 3D face reconstruction problem. This algorithm also requires multiple views of the same face in order to build the 3D model. Since BA is usually employed as a final refinement method for 3D reconstructions, we need to provide it with an initial face model and pose estimation for the different views. In order to design a generic method, the mean shape of the BFM will be used to initialize the BA fitting of any input face, and initial pose estimates of every view will be achieved using POSIT with the mean shape of the BFM.

3.2.7 Free Deformations

Free deformations lay out a non-rigid model that is adjusted according to the 2D observations of corresponding points. We refer to this approach as 'free deformations' because it is not based on back-projections from a frontal view, does not treat the model as a rigid one like the scaling approach, the deformation is not constrained by an orthonormal subspace like in the PCA approach, and deformation is not guided by a specific algorithm like in the BA approach. We present three different approaches in this category. The first one clusters the facial points based on the facial region they belong to according to heights in the face, and deforms these clusters along the vertical axis; the second one starts from a generic face and only adjusts the depth components; the third one considers every facial point independent and freely deforms them to find the fitting that best accounts for the image observations. The three methods make

use of different views of the head to fit the model, following the framework laid out in Algorithm 1.

3.2.7.1 Region-Based Deformation

There are two important motives that have led to proposing this region-based deformation in the manner it will be explained right after. On the one hand, we know that human faces show a significant symmetry with respect to the vertical axis in the *XY* plane, which may lead us to think that finding an accurate fitting for the 3D model points along the *X* axis is not very important, since image variations of symmetric points due to pose changes can compensate each other in terms of estimating the rotation correctly. On the other hand, theory of GEMs [63], [64] claims that depth information in a face (the *Z* component) is not significantly discriminative among different individuals and a generic depth map can be used to model any face.

Based on those two assumptions, we propose a region-based deformation that deforms the face along the *Y* spatial component. The facial points are clustered according to their vertical height in the face: points in the eyebrows, points in the eyes, points in the nose and points in the mouth are the

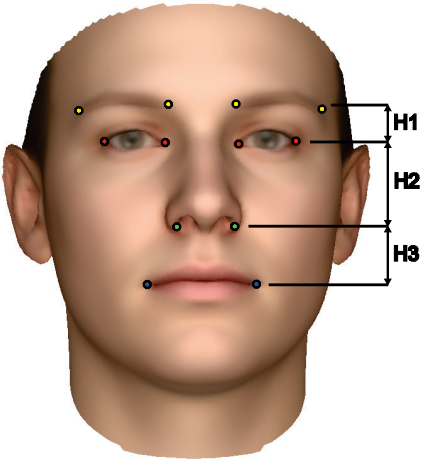

Figure 3.5: Region-based deformation. Relative heights, H1, H2 and H3, between eyebrows cluster (yellow), eyes cluster (red), nose cluster (green) and mouth cluster (blue) are adjusted.

four clusters. Starting from the generic BFM and assuming the height of one of the groups is fixed, three relative heights need to be adjusted based on the 2D observations, as shown in Fig. 3.5. The relative heights are measured averaging the height of all the points in each of the clusters. The multi-view based optimization framework from Algorithm 1 is thus initialized with the generic heights of the mean shape of the BFM, and the resulting 3D shape is reconstructed at each iteration by adjusting each point's height based on the variation of the relative height of its cluster. As already explained, X and Z spatial components of the facial points are always those of the generic model.

3.2.7.2 Depth Fitting

The depth fitting approach consists in just deforming the depths of the points, leaving X and Y components unaltered. This is the opposite approach as the one proposed by GEMs. The latter claims that depth information in a face is not significantly discriminative among different individuals and a generic depth map can be used to model any face. However, it is still unclear whether the GEM's assumption about depths and all its implications are a valid approach in any scenario since, for instance, it is derived from it that depth information in faces is not significantly discriminative for modeling image appearance variation due to 3D pose changes.

In order to analyze the validity of this assumption, we propose the opposite approach, in which X and Y components are assumed not to be discriminative among individuals and are approximated by a generic face model, i.e. the mean shape of the BFM. The depths of the facial points are fitted to the 2D observations using the multi-view optimization framework described in Algorithm 1.

3.2.7.3 Fully-Free Deformation

Finally, a fully-free deformation is proposed, in which each spatial component of each facial point is assumed to be completely independent. This means that each point can vary freely during the fitting process. The model is built based on the multi-view optimization framework from Algorithm 1. The mean shape of the BFM is used to initialize the fitting, and X, Y and Z components of all the facial points are iteratively adjusted until the reprojection error is minimized.

This method is similar to BA in Section 3.2.6, in which all the points are also adjusted independently. The main conceptual difference between both methods is that, whereas the fully-free deformation considers multiple head poses on a single camera, the BA addresses the problem as a single head viewed from different cameras. In addition to that, the fully-free deformation follows Algorithm 1 for a reprojection error minimization based on the Nelder-Mead simplex method [70], whereas BA works with a LM minimization scheme.

3.3 Fitting Results

This section is devoted to presenting the fitting results of each of the approaches described, and provides a performance comparison. First, the framework on which the 3D face reconstruction has been carried out and model fitting methods have been evaluated and compared is described. Then, the results are presented and discussed in order to find the most suitable face reconstruction method.

3.3.1 Framework

All the model fitting approaches presented in the previous section have been evaluated in the same conditions. The synthetic head pose database presented in Chapter 2 has been used for face reconstruction and evaluation, since it is a controlled environment in which all the variables are manipulated at free will. The advantages of using this environment are thus numerous: 1) the real 3D faces are available as ground truth for measuring the reconstruction error; 2) the exact anatomical corresponding points among faces are known; 3) noise-free 3D ground truth for HPE evaluation is available; 4) noise-free 2D image ground truth for face reconstruction is available, avoiding tracking errors affecting the face reconstruction performance evaluation; 5) ideal camera parameters are known so that they do not affect the evaluation either; 6) the database contains 36,000 total frames, 3,600 per user, a number large enough for good evaluation of model fitting approaches.

Model fitting approaches using back-projection from frontal views (i.e. Sections 3.2.1, 3.2.3.1 and 3.2.5) make use of the initial frame of the twelve videos available for each user. It is important to remind that the head pose ground truth in the synthetic database copies the one in the UPNA real database presented in Section 2.1, and therefore the head pose at each initial frame will probably differ

slightly from a complete frontal view of the face. In order to compensate for this misalignment, the calculated back-projections are averaged across all the initial frames for each face. This procedure is in accordance with a real application needing a frontal view of the user for face reconstruction, where the user would be asked to try to stay still facing the camera for a certain amount of time, while all the frames acquired were averaged in order to obtain a more robust result.

In the case of multi-view based fitting approaches (i.e. Sections 3.2.3.2, 3.2.4, 3.2.5.3, 3.2.6 and 3.2.7), one frame per second from each video (excluding pure-translation videos) has been selected for the reconstruction, adding up to 90 frames per face and assuring a certain variety of head poses. Exactly the same video frames have thus been used for reconstruction in all the multi-view based methods, which guarantees a fair performance comparison.

The twelve facial points of interest have been annotated in 3D in all the synthetic faces, since the exact anatomical correspondences between faces are known and therefore also their exact 2D projections in each frame of the database. These real 3D shapes can thus be compared with the reconstructed shapes in order to obtain a 3D reconstruction error. Besides, HPE for the complete database has been carried out using POSIT with the reconstructed 3D shapes and the 2D ground truth, in order to analyze the impact of face reconstruction in the subsequent HPE output. Errors for HPE are obtained by comparing these estimations with the 3D ground truth of the database.

Using the 2D ground truth for both the face reconstruction and the subsequent HPE implies that the reconstruction methods are evaluated and compared in perfect tracking conditions. This is an interesting way of comparing model fitting approaches because we avoid other variables affecting the reconstruction. However, it has also been considered of interest to study their performance in noisy environments. For this purpose, Gaussian noise of zero mean and selected values of standard deviation has been introduced in the 2D ground truth simulating tracking inaccuracies for both the face reconstruction and the subsequent HPE. The tracking noise is intended to affect both coordinates of the 2D images and thus follows a bivariate normal distribution, whose probability density function (PDF) is

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left(\frac{(x-\mu_x)^2}{\sigma_x{}^2}+\frac{(y-\mu_y)^2}{\sigma_y{}^2}-\frac{2\rho(x-\mu_x)\left((y-\mu_y)\right)}{\sigma_x\sigma_y}\right)} \quad (3.14)$$

where μ_x and μ_y are the mean of the two image coordinates, σ_x and σ_y the standard deviations and ρ is the correlation value between both dimensions. Assuming they are uncorrelated, and making $\mu_x = \mu_y = 0$ and $\sigma_x = \sigma_y = \sigma$, the PDF results in

$$f(x,y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.15)$$

The expected value of a measurable function of x and y, $\phi(x,y)$, is defined as

$$E[\phi(x,y)] = \int_{-\infty}^{\infty} \phi(x,y) f(x,y) dxdy \quad (3.16)$$

We can then calculate the relation between the standard deviation σ of the random noise applied to the 2D ground truth and the average tracking error we may expect, measured as the average Euclidean distance between the 2D ground truth and the noisy 2D points. The measurable function is then in this case the Euclidean distance $\phi(x,y) = \sqrt{x^2+y^2}$, and equation (3.16) can be rewritten as

$$E[\phi(x,y)] = \int_{-\infty}^{\infty} \sqrt{x^2+y^2} \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} dxdy \quad (3.17)$$

Solving equation (3.17), we obtain that

$$E[\phi(x,y)] = \sigma\sqrt{\frac{\pi}{2}} \quad (3.18)$$

where $E[\phi(x,y)]$ is the expected tracking error if we introduce a 2D random noise of mean zero and standard deviation σ.

Using all the above, the model fitting methods that perform best in perfect tracking conditions have been selected and tested in noisy tracking conditions, specifically for noise levels (average Euclidean error) of 1, 2, 3 and 4 pixels. These noise levels have been applied to every frame of the synthetic database, creating a sort of 'noisy databases', where the selected model fitting approaches have been evaluated following the same procedure as for the perfect tracking database. Results using the real 3D model have been included as reference in the performance comparison.

3.3.2 Results and Discussion

First of all, it should be clarified that, in some tables and figures of this section, a code to identify each model fitting approach is used in order to save space. This code refers to the subsection, inside Section 3.2, in which the approach has been described (e.g. full deformation approach is described in Section 3.2.7.3 so it is referenced with the code '7.3'). Note that the real 3D model is shown as reference among the approaches with the code '0'.

This section is divided in two parts: first, fitting results in perfect tracking conditions for all the methods proposed will be presented and discussed. Then, the methods that show the best performance will be selected and their performance in noisy tracking conditions will be presented and discussed.

3.3.2.1 Perfect Tracking Conditions

Model fitting 3D errors for all the reconstruction approaches proposed are presented in Table 3.1 and Fig. 3.6. The fitting approaches in the bar chart are identified by their codes. The first error shown corresponds to the reconstructed 3D face directly compared to the real 3D model, whereas the second error results from comparing the shapes after registration through Procrustes analysis. Procrustes analysis has been applied in order to obtain the translation, rotation and scale that best adjusts the reconstructed shape to the real one. The aim of this process is to have a comparison metric that only depends on the 3D relative position of the points in the face. That is, a comparison of the internal configuration of the reconstructed and real 3D shapes, after errors due to difference between origins, misalignment of the coordinate systems, or global scale are corrected.

In general, results comparing registered and non-registered reconstructions are coherent and an improvement of around 30% is achieved due

Table 3.1: Model fitting 3D errors, measured as point-to-point Euclidean distance for the 12-point facial feature model. Errors for the reconstructed shapes with and without registration to the real 3D shape are shown.

Fitting Approach		Fitting Error (mm)	
		Non-registered	Registered
0.	Real model	0,00 ± 0,00	0,00 ± 0,00
1.	Cylindrical model	7,17 ± 3,17	5,79 ± 3,43
2.	Generic BFM	4,81 ± 2,32	3,10 ± 1,53
3.1.	Scaled – aspect ratio	5,03 ± 2,26	3,40 ± 1,48
3.2.	Scaled – full-scaling	6,15 ± 2,78	2,85 ± 1,14
4.	PCA-based	4,58 ± 2,01	1,56 ± 1,08
5.1.	Back-proj. – generic depths	4,62 ± 2,41	2,94 ± 1,56
5.2.	Back-proj. – depth interpolation	4,57 ± 2,36	2,78 ± 1,58
5.3.	Back-proj. – multiple view	4,17 ± 1,90	2,48 ± 0,94
6.	Bundle adjustment	2,76 ± 1,09	0,00 ± 0,00
7.1.	Free def. – region-based	4,73 ± 2,31	2,95 ± 1,34
7.2.	Free def. – depth fitting	4,90 ± 1,92	3,15 ± 1,10
7.3.	Free def. – fully-free	4,28 ± 1,81	1,37 ± 0,53

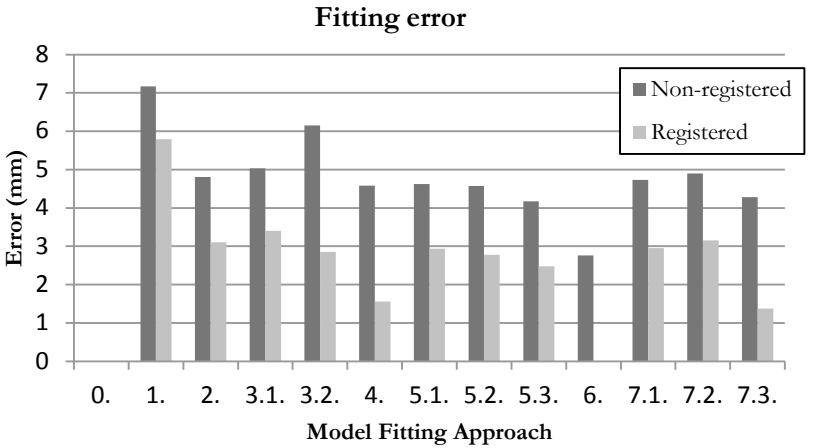

Figure 3.6: Bar chart of the model fitting 3D errors. Errors for the reconstructed shapes with and without registration to the real 3D shape are shown.

to the registration process. Non-registered errors are mainly around 4-5mm and registered ones around 3mm. There are, however, a few cases where this improvement is still greater: the full-scaling approach (3.2), the PCA-based approach (4), BA (6) and the fully-free deformation (7.3). The last three actually achieve the best fitting results among all the methods after registration, especially

BA, which practically reaches a null error, meaning that the internal configuration of the reconstructed shape is identical to the real model.

The full-scaling approach probably adjusts the scales to reach the best internal configuration that accounts for the 2D observations at the expense of suffering from a poor global scale. This statement is supported by translation and rotation errors in Table 3.2 – Fig. 3.7 and Table 3.3 – Fig. 3.8 respectively. While 3D fitting and translation errors of the full-scaling approach are quite high in comparison with other methods, rotation error is the 5th from the 13 total approaches proposed. The global scale of the reconstruction will clearly affect the 3D fitting error and the translation error in HPE, whereas it may have no effect on the rotation error. Let's explain this with a simple example. Imagine having a downscaled version of a real face. Any 2D observation of the real face can be accounted for using the downscaled version, just by posing it closer to the camera. Finding the global scale of the model based only on 2D observations is in fact a great challenge, since there will always be an indetermination. This problem is nowadays overcome using devices such as depth cameras (e.g. Kinect), currently at competitive prices, which makes them more and more employed for this kind of applications.

It should also be pointed out that approaches based on multi-view optimization (3.2, 4, 5.3, 6, 7.1, 7.2, and 7.3) perform generally better than approaches based on a single view of the face (1, 3.1, 5.1 and 5.2). The CHM approach (1) is actually the worst method in all the comparison metrics, which leads us to conclude that approximating the face by a cylindrical surface is not a good approach. In the case of using a generic face, the BFM (2), it shows a performance far from the best but also better than the worst approaches to a certain extent. It seems an interesting approach because it will always provide a stable performance in any environment, it does not require a model fitting process, and the results may be acceptable depending on the target application.

The importance of depth estimation of 3D points is also an issue worth discussing. It has already been mentioned in this work, in Section 3.1, that GEMs [63], [64] claim that depth information in a face is not significantly discriminative among different individuals and, consequently, the depths of a set of facial points could be approximated by the depths of the same anatomical points obtained from a generic 3D face model. In order to assess this statement, we should pay attention to some of the methods presented. The approach 5.1, back-projection based fitting using generic depths, is purely a GEM approach, whereas the approach 7.2, free depth deformation using generic X and Y components, could

Table 3.2: Translation errors for the proposed model fitting approaches.

Fitting Approach		Translation Error (mm)			
		T_X	T_Y	T_Z	Avg
0.	Real model	0,11	0,04	0,89	0,35 ± 0,21
1.	Cylindrical model	10,24	8,29	3,75	7,42 ± 5,73
2.	Generic BFM	6,90	5,78	2,96	5,21 ± 4,06
3.1.	Scaled – aspect ratio	7,21	5,81	3,00	5,34 ± 4,17
3.2.	Scaled – full-scaling	9,13	6,09	3,93	6,38 ± 5,17
4.	PCA-based	7,97	5,60	2,82	5,46 ± 4,34
5.1.	Back-proj. – generic depths	8,13	6,15	3,64	5,97 ± 4,74
5.2.	Back-proj. – depth interpolation	6,26	4,66	3,48	4,80 ± 3,82
5.3.	Back-proj. – multiple view	6,82	4,24	3,13	4,73 ± 3,78
6.	Bundle adjustment	3,11	2,11	1,39	2,20 ± 1,73
7.1.	Free def. – region-based	6,76	5,66	2,85	5,09 ± 3,96
7.2.	Free def. – depth fitting	4,23	2,43	2,68	3,11 ± 2,50
7.3.	Free def. – fully-free	4,80	3,59	2,45	3,61 ± 2,83

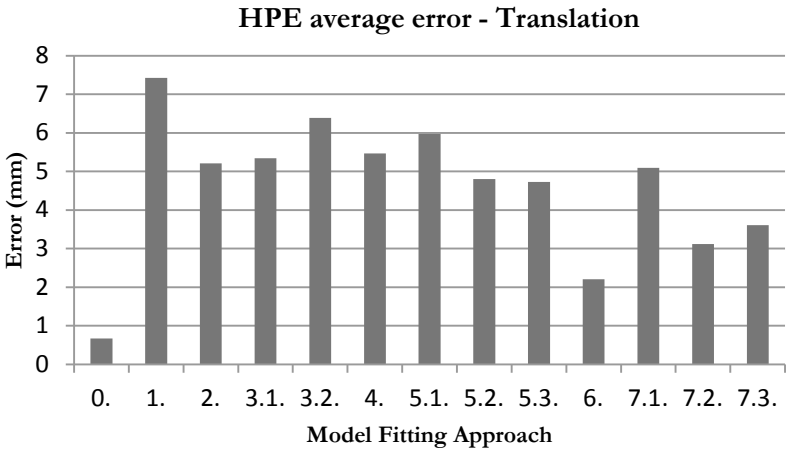

Figure 3.7: Bar chart of translation errors for the proposed model fitting approaches.

actually be considered as the opposite approach. It is interesting to note that, whereas the GEM achieves similar but slightly better 3D model fitting results, the opposite approach shows a significantly better performance when using the reconstructed shape for HPE. This leads us to claim, answering a question posed by the recent GEM approaches, that, by using an elastically deformed generic depth map, we cannot reliably account for 2D variations due to 3D pose changes. The significance of accurately estimating depths is also supported by the poor

Table 3.3: Rotation errors for the proposed model fitting approaches.

Fitting Approach		Rotation Error (°)			
		Roll	Yaw	Pitch	Avg
0.	Real model	0,02	0,07	0,02	0,04 ± 0,03
1.	Cylindrical model	0,23	1,19	0,98	0,80 ± 0,61
2.	Generic BFM	0,16	0,94	0,75	0,62 ± 0,48
3.1.	Scaled – aspect ratio	0,16	0,95	0,75	0,62 ± 0,49
3.2.	Scaled – full-scaling	0,15	0,51	0,34	0,33 ± 0,27
4.	PCA-based	0,10	0,55	0,39	0,35 ± 0,27
5.1.	Back-proj. – generic depths	0,24	0,98	0,75	0,66 ± 0,52
5.2.	Back-proj. – depth interpolation	0,24	0,86	0,63	0,57 ± 0,45
5.3.	Back-proj. – multiple view	0,22	0,51	0,27	0,33 ± 0,26
6.	Bundle adjustment	0,03	0,09	0,04	0,05 ± 0,04
7.1.	Free def. – region-based	0,16	0,93	0,74	0,61 ± 0,48
7.2.	Free def. – depth fitting	0,14	0,27	0,18	0,20 ± 0,16
7.3.	Free def. – fully-free	0,07	0,36	0,24	0,23 ± 0,18

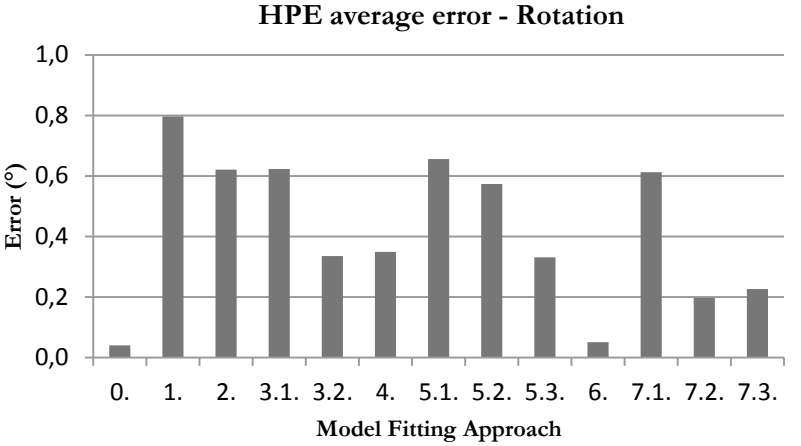

Figure 3.8: Bar chart of rotation errors for the proposed model fitting approaches.

results shown by the CHM, which not only uses a generic depth map but approximates a human face surface by a cylinder. In fact, approaches 2 and 5.1 use the same X and Y coordinates in the built shape, and only differ in the depths' approximation (cylinder or human). Another result supporting this claim is the comparison between the back-projection based fitting methods (5.1, 5.2 and 5.3). The first approach uses a generic depth map (5.1) and achieves the worst results among the three. The second approach (5.2) interpolates the depths

Table 3.4: Computation time for the proposed model fitting approaches.

	Fitting Approach	Computation Time (s)
0.	Real model	0,00
1.	Cylindrical model	0,00
2.	Generic BFM	0,04
3.1.	Scaled – aspect ratio	0,00
3.2.	Scaled – full-scaling	10,29
4.	PCA-based	1204,20
5.1.	Back-proj. – generic depths	0,00
5.2.	Back-proj. – depth interpolation	0,03
5.3.	Back-proj. – multiple view	73,09
6.	Bundle adjustment	28,17
7.1.	Free def. – region-based	10,71
7.2.	Free def. – depth fitting	56,63
7.3.	Free def. – fully-free	583,82

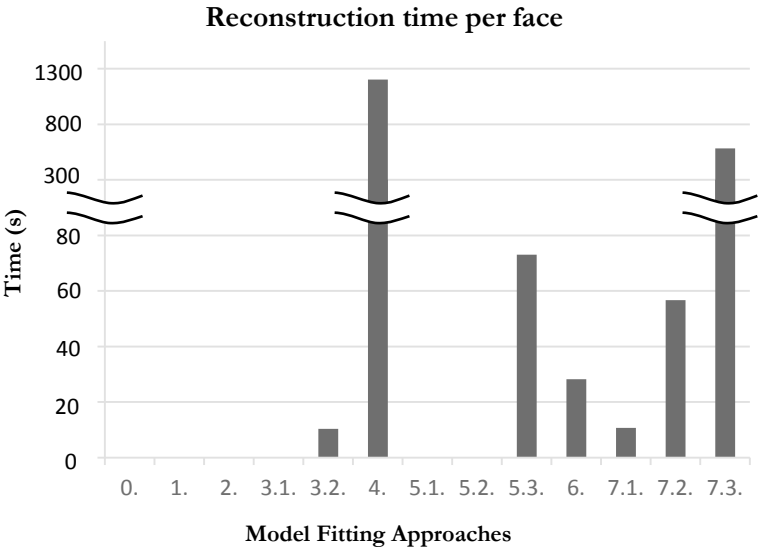

Figure 3.9: Bar chart of computation time for the proposed model fitting approaches.

from a generic depth map based on the back-projections, and shows a slight improvement in performance. The third approach (5.3), however, calculates the depth through a multi-view based optimization process, fitting the depth of each of the 3D points. While the improvement in 3D fitting error and translation error is also slight, it is clearly more noticeable in rotation error. Therefore, an accurate estimation of point depths seems to be critical for achieving good rotation estimation in HPE.

Average reconstruction times have also been calculated and are shown in Table 3.4 and Fig. 3.9. The reconstruction has been performed using Matlab on an IntelCore i5 PC with 6GB of RAM. The first thing to point out is that the PCA-based reconstruction is, by far, the slowest one, followed by the fully-free deformation. The two approaches are in the range of several minutes (20 and 10 approximately), whereas the rest of the approaches are in the range of seconds. Remember that our PCA-based method optimizes 75 of the 199 principal components of the BFM. The computation time could be reduced by half optimizing only 35-40 modes (remember the linear tendency in Fig. 3.4f). However, that would be at the expense of losing accuracy in both reconstruction and HPE. In any case, the PCA-based model fitting method does not seem a suitable option, because it cannot reach competitive results with a reasonable reconstruction time.

Apart from the two slowest methods mentioned, it should be noticed that, among the rest, single-view based methods reconstruct the face significantly faster (in the range of milliseconds) than multiple-view based methods (in the range of seconds), as it could be expected. This is, nevertheless, at the expense of losing accuracy, especially in rotation estimation. Single-view based methods can thus be a suitable option when reconstruction time requirements are critical for the application and accuracy requirements are looser.

The best compromise is achieved, without doubt, by the BA. It is by far the most accurate model fitting method and employs in average 28 seconds to reconstruct each face. Besides, we can observe that its inherent problem of the need of a good initialization is overcome using the generic BFM for the initial estimate. BA is followed in performance by the fully-free deformation and the depth fitting using generic X and Y components, although the former presents the problem of the long reconstruction time required, and the latter does not achieve very good 3D fitting results but performs very well in HPE.

3.3.2.2 Noisy Tracking Conditions

Based on the results described above, the following methods have been included in the performance comparison in noisy tracking conditions: the real 3D model (0), the mean shape of the BFM (2), the full scaling approach (3.2), the back-projection with multi-view depth fitting approach (5.3), BA (6), depth fitting with generic X and Y components (7.2), and the fully-free deformation (7.3). The first two have been selected as references: the real model, and a generic model that does not depend on the input face. They do not go under any deformation and thus are not affected by the tracking noise. The rest of the methods have been selected in order to have all the main groups of approaches represented, choosing the most accurate option from each. The CHM has been discarded because of its poor accuracy results, and the PCA-based method because of its large computation time. All in all, the methods included in this subsection are possibly the best options for any application.

Model fitting error in 3D has been measured again for non-registered (Fig. 3.10a) and registered (Fig. 3.10b) reconstructed 3D shapes, in different tracking noise level conditions. The first noise level in the figures, a 0-pixel noise, corresponds to the perfect tracking setup presented in the previous section. The real 3D shape shows obviously a null fitting error for any noise level. The generic model, the mean shape of the BFM, shows a constant error independent from the tracking noise because it is not really a reconstruction, but a fixed generic model, therefore completely robust against noise.

Regarding the rest of the methods, when their reconstruction is not registered to the real model, the fitting error does not always show a clear tendency against tracking noise (Fig. 3.10a). There seems to be an ascending tendency of the error in the case of the free deformations (7.2 and 7.3), a more notorious one in the case of the back-projection with multi-view depth fitting approach (5.3), and no tendency in the full-scaling approach (3.2) and the BA (6). However, this is once again likely to be due to a global scaling factor, since a clear tendency is observed when looking at the fitting errors after registering the reconstructed faces to the real ones (Fig. 3.10b). In this figure, every fitting method shows a poorer reconstruction as the noise in the 2D observations increases. The problem of not knowing the global scale of the face cannot be solved using only 2D views of the face, and it clearly masks fitting errors due to the internal configuration of the facial points.

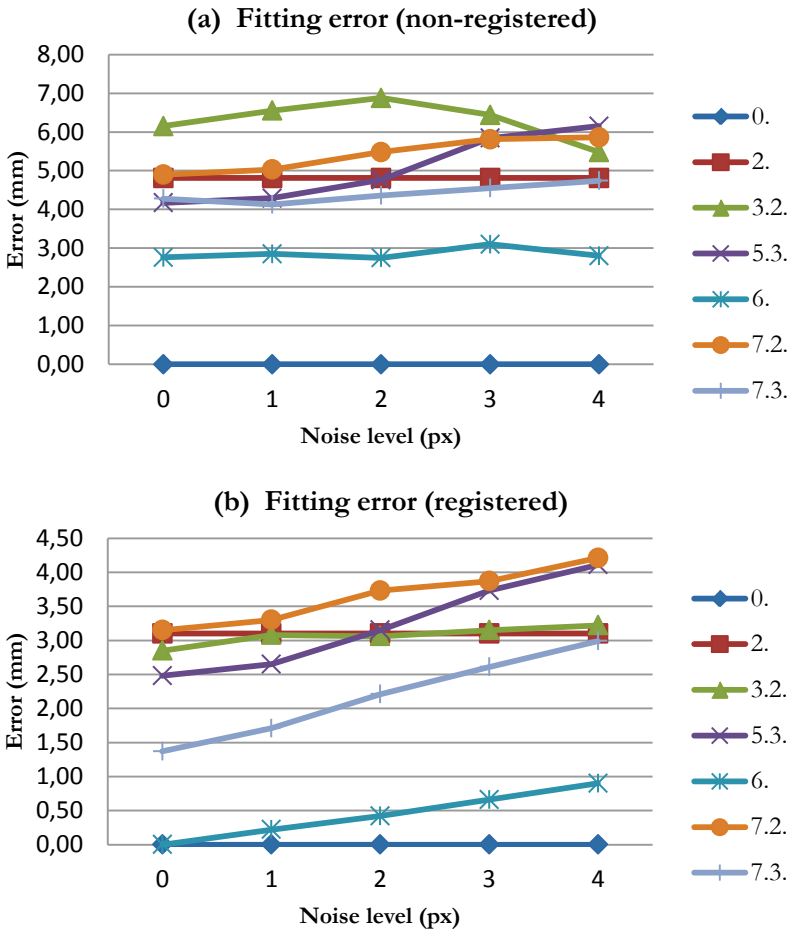

Figure 3.10: Model fitting 3D errors when tracking noise is present in the reconstruction. (a) Non-registered reconstruction. (b) Reconstruction registered to the real 3D shape.

In general, we can observe that the BA clearly obtains the best fitting results under any noise level (always under 1mm after registration and around 3mm without), followed by the fully-free deformation. The latter obtains similar results as the generic BFM for the highest noise level. The full-scaling approach seems only valid after registration, but still does not show a significant improvement from using the generic BFM and even performs worse in high noise conditions. Approach 5.3 is also outperformed by the generic model after

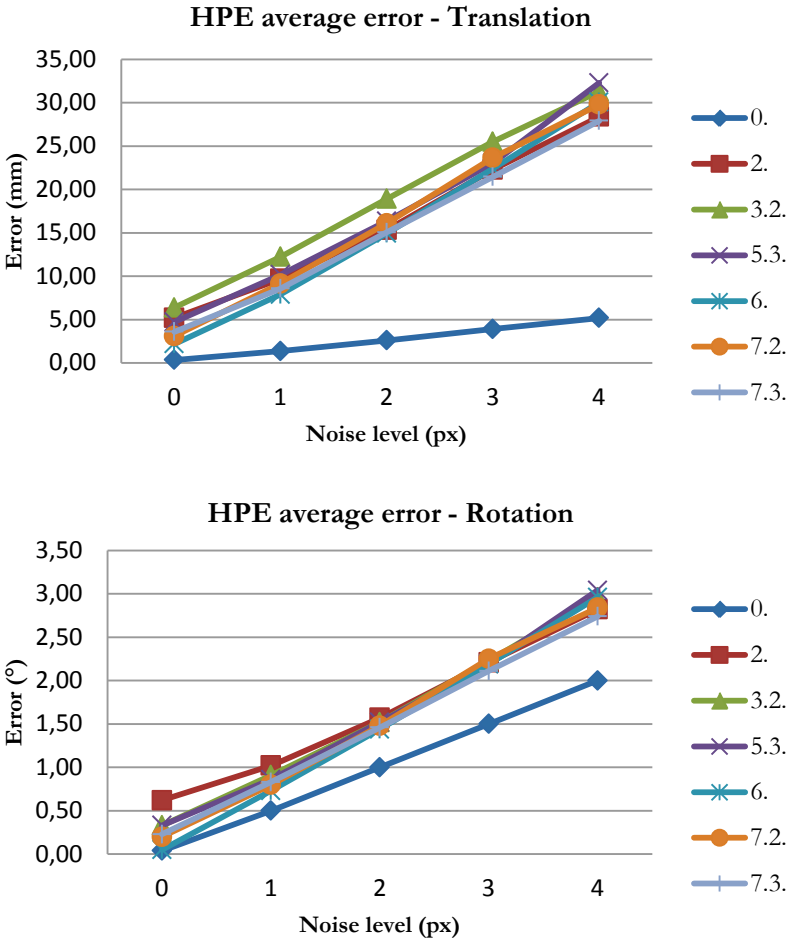

Figure 3.11: Incremental HPE error when tracking noise is present in the reconstruction and subsequent HPE evaluation.

2-pixel noise, and the case of method 7.2 is special because it achieves a worse fitting than the generic model for any noise level. This is striking because it should be reminded that approach 7.2 uses generic X and Y components and only fits the depths based on multiple 2D views of the face. We will get back to this later in this discussion.

Using the reconstructed faces, HPE is carried out on the whole database and translation (Fig. 3.11a) and rotation (Fig. 3.11b) errors are presented. It is important to remember that the noise, of the corresponding level, is present both in the 3D shape reconstruction process and the subsequent HPE evaluation. Both HPE errors (translation and rotation) show a clear difference between the performance of the real 3D model and the rest, which becomes more and more significant as the noise level increases. When no noise is present, the HPE performance of the different methods can be compared, as it has been done in the previous section. Nevertheless, in the presence of noise, the curves tend to overlap and their differences become indistinguishable.

There are two main reasons that explain this observation: 1) the lack of registration of the reconstructed face; 2) the tracking noise affecting the calculation of the relative HPE error. Let's explain this in detail. We have seen that not having the reference of the real 3D face in order to register the reconstructed shape affects significantly the fitting error, especially due to the global scaling factor. This will also affect the estimation of translation in HPE. There can also be a misalignment between the coordinate system of the real model and the coordinate system of the reconstructed shape, which will affect the estimation of rotation in HPE. This ignorance of the real model, with which the ground truth for HPE has been obtained, is why pose estimation errors are usually measured relatively, in an incremental way from frame to frame, and not in an absolute way. And, in this case, the relative HPE error has been obtained by aligning the estimation and the ground truth for the initial frame of each video sequence in the database (we will discuss further the meaning of this in the next chapter). The problem is that, when noise is present in the 2D points, the estimation used in the calculation of the transformation for misalignment correction will be incorrect. This effect is more evident as the noise level increases. Moreover, the fact that the noise is of random nature enhances the effect, since the noisy 2D configuration of the initial frame is totally independent from the noisy 2D configurations of the rest of the frames. This last inconvenient is partially avoided in real tracking situations, because tracking errors usually show a drifting effect, more similar to an offset, that may be compensated to some extent by measuring incremental errors.

In order to prove all the above, two paths have been followed regarding HPE error calculation: 1) carry out HPE after registering the reconstructed 3D shape to the real face; 2) perform a noise-free alignment to calculate the relative HPE errors. Translation and rotation errors obtained using the registered 3D

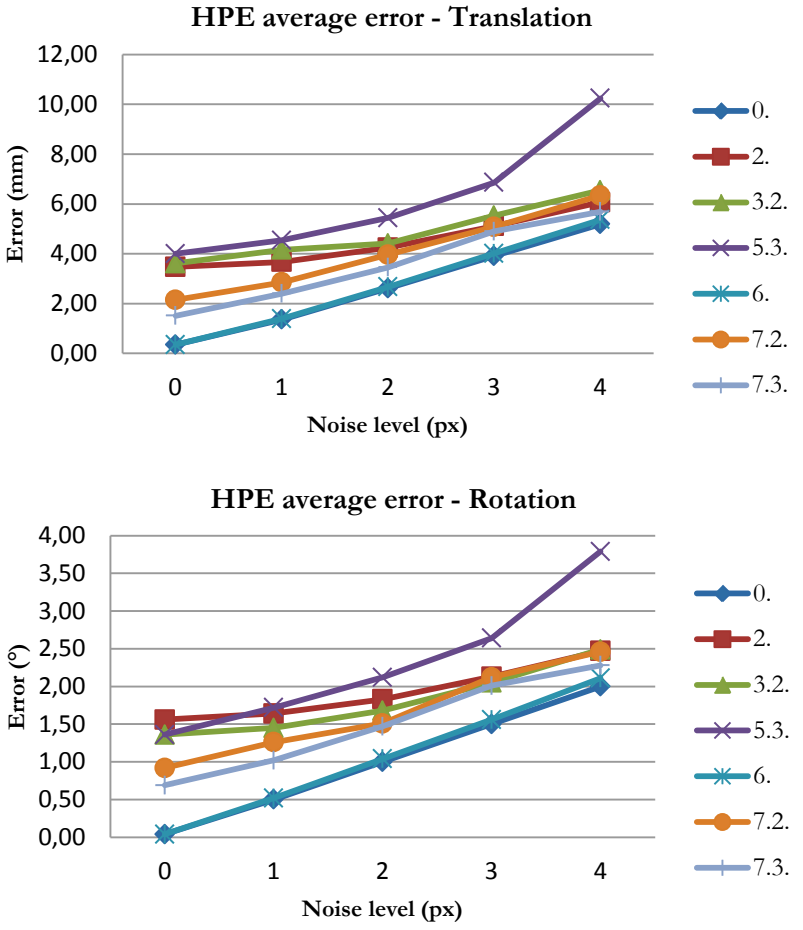

Figure 3.12: HPE error when tracking noise is present in the reconstruction and subsequent HPE evaluation, previous registration of the reconstructed 3D shapes to the real faces.

shapes are shown in Fig. 3.12. As already explained, registration through Procrustes analysis allows us to obtain a reconstructed 3D shape that only differs from the real one in its internal configuration, but not in global scale, translation or rotation. This way, a perfect alignment between HPE estimation and ground truth reference systems is achieved, and HPE errors will only be due to differences in that internal configuration of the 3D points. Fig. 3.12 shows that the differences between the different approaches are now clearer, and the

difference with respect to HPE results using the real model is significantly reduced. In fact, the registered BA model shows very similar results to the real model, showing just a slightly worse performance as the noise level increases. Approach 7.3 is the next best one, although the difference with respect to the BA and the real model is more significant, and it is followed closely by approach 7.2. The generic BFM and the full-scaling approach show quite similar results, the scaling deformation proving useful for rotation estimation but slightly worsening translation estimation.

It is still true that the curves tend to converge toward the same HPE accuracy level as the tracking noise increases. This leads us to conclude that the HPE error caused by tracking inaccuracies prevails over the HPE error introduced by 3D model inaccuracies. The curve of approach 5.3, however, clearly diverges from the rest, the HPE error increasing faster as the noise level is incremented. This a priori strange behavior has a simple explanation: it is the only method in this comparison that is based on a back-projection of 2D points from a frontal view of the face (to estimate the X and Y coordinates of the 3D points). As the noise in 2D increases, these back-projections will become more inaccurate, leading to a poor reconstruction. Since the rest of the methods in this comparison are based on multiple views of the face, they will be able to compensate to some extent the increasing noise level and will therefore be less affected by it.

In order to show the effect of the global scale of the model in HPE, registration through Procrustes analysis without the scaling factor has also been carried out between the reconstructed shapes and the real models. Results are shown in Fig. 3.13. It can be observed that rotation errors remain unaffected by the global scale, whereas it results critical for translation estimation. Translation error is in this case totally dominated by the global scale and not by the 2D noise level.

Finally, HPE results after noise-free alignment between estimation and ground truth are shown in Fig. 3.14. The face reconstruction and HPE processes are carried out similarly, using the noisy image sets from each noise level, but the transformation to align the estimation and ground truth of head pose for the initial frame of each sequence has been calculated using noise-free 2D point sets. This approach is very interesting because it is quite in accordance with reality: we do not need to know the real 3D model, since there is no registration of the reconstructed shape, but we assume that we have a noise-free initial detection of the facial points in each image sequence. This could be implemented in a real application by asking the user to face the camera for a very short period of time

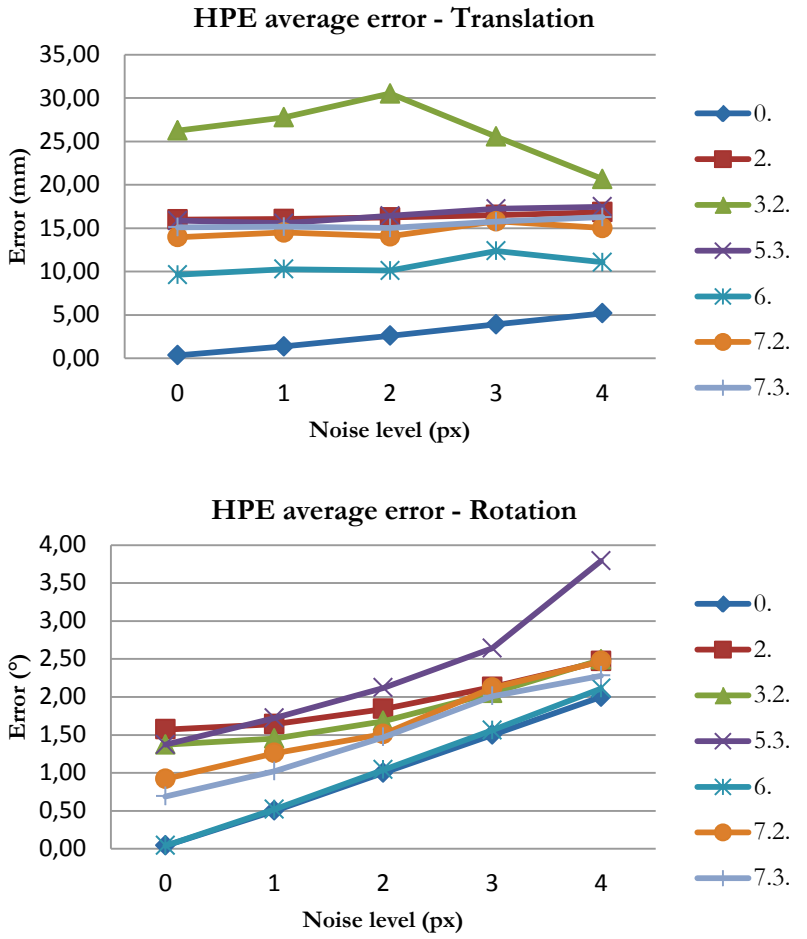

Figure 3.13: HPE error when tracking noise is present in the reconstruction and subsequent HPE evaluation, previous registration of the reconstructed 3D shapes to the real faces without varying the global scale.

while several frames were acquired and the detected points averaged in order to obtain a stable and noise free estimation. The transformation matrix that aligns the estimation and ground truth would then be calculated and applied to any subsequent frame of the sequence, which could of course contain noisy tracked points, as in the case of the simulation with the noisy datasets presented here.

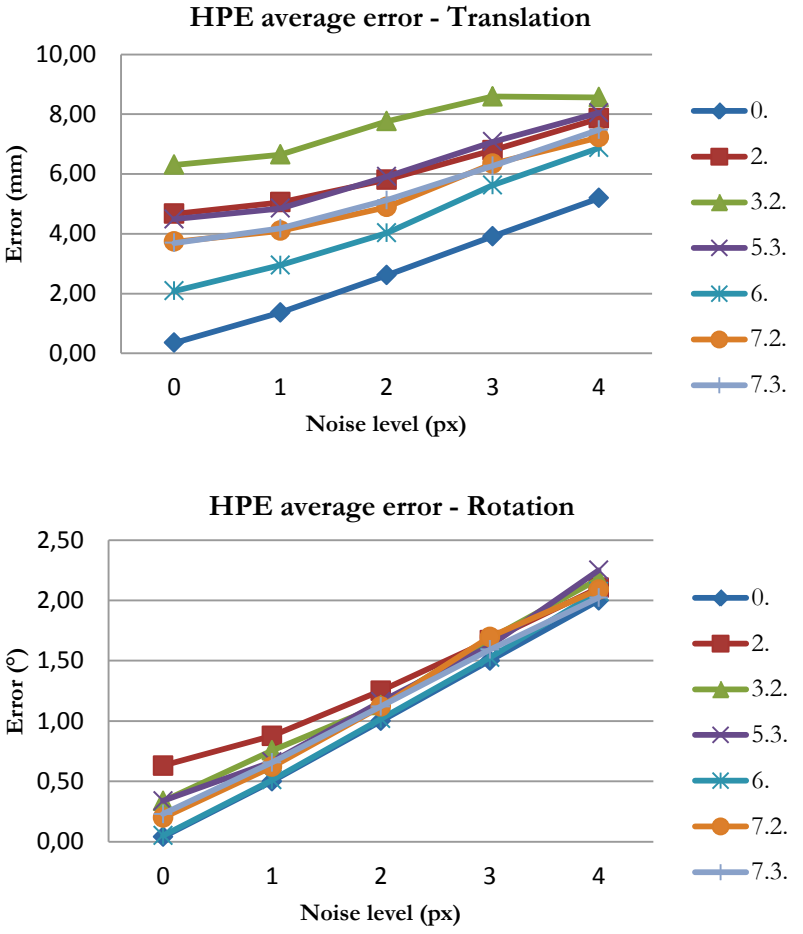

Figure 3.14: Incremental HPE error when tracking noise is present in the reconstruction and subsequent HPE evaluation, after noise-free alignment of the estimation and ground truth for the initial frame of each sequence.

These results are quite in accordance with the ones obtained after registering the reconstructed faces to the real ones. Regarding the rotation, using a generic model is the worst option for low noise levels but, as the noise increases and the rotation error with it, all the curves tend to converge to a same accuracy level. That is, the tracking noise tends to prevail over 3D model inaccuracies. This is even true for the case of using the real 3D model. The BA is once again the

best option, at least for low tracking noise, where the curves are more distinguishable. Regarding the translation, we can observe also a convergence tendency between the different fitting approaches as the noise increases, but this time the real model remains over 2mm ahead of the rest in accuracy. This behavior is probably due to global scaling, which cannot be totally corrected through the noise-free alignment, at least not as well as through registration of 3D shapes. This is also evident for the full-scaling approach, which shows the poorest performance for translation estimation, worse than the generic model, which keeps the global scale closer to the real one. The last thing worth mentioning is that approach 5.3, based on back-projections with multi-view depth fitting, improves its performance compared to results after registration. The most likely reason is that the noise-free alignment is able to compensate for the noisy back-projections regarding the HPE better than the registration of the 3D shapes. The inaccurate back-projections will produce a systematic HPE error that may at least partially be corrected by the transformation calculated using noise-free 2D points.

3.4 Full 3D Face Reconstruction

We present in this section a method to recover a dense full-head 3D model from the sparse 3D shape reconstructed through one of the approaches presented above. The full reconstruction is then evaluated and the results are discussed.

3.4.1 Methods

The parametric face model presented in Section 3.2.4 consists of 53,490 vertices representing a dense 3D shape of the head, described by equation (3.10). We could therefore follow a similar PCA-based fitting approach in order to adjust the dense 3D mesh to a sparse reconstruction. The procedure is as follows:

1) BA is used to reconstruct a sparse 3D shape of 12 points, as described in Section 3.2.6. This approach is selected among all because it achieves the best sparse reconstruction according to Fig. 3.6, almost perfect after registration to the real 3D shape.

2) Following equation (3.11), the optimal coefficients $\hat{\boldsymbol{\alpha}}$ that best adjust the 12 facial points $\boldsymbol{X}_f$ in the BFM to the reconstructed sparse 3D shape are found. It

is important to note that this optimization process minimizes the distance between both point clouds in 3D, as opposed to the 2D minimization of projections carried out in Section 3.2. A regularization term may be introduced in order to constrain the shape deformation as in equation (3.12), resulting in a 3D Euclidean error metric e defined as

$$e = \frac{1}{F}\sum_{f=1}^{F}\sqrt{\left(x_f - x_f{}'\right)^2 + \left(y_f - y_f{}'\right)^2 + \left(z_f - z_f{}'\right)^2} + a\frac{1}{L}\sum_{l=1}^{L}{\alpha_l}^2 \qquad (3.19)$$

3) The optimal coefficients $\hat{\boldsymbol{\alpha}}$ resulting from the optimization are introduced in equation (3.10) to recover the dense 3D shape $\boldsymbol{s}(\hat{\boldsymbol{\alpha}})$. Note that just the shape is recovered and not the texture, since no texture information is being used.

The reconstructed dense 3D shape may then be compared with the complete real model of each synthetic user in order to compute full-head 3D reconstruction errors.

3.4.2 Results and Discussion

Table 3.5 presents the 3D errors measured as the average point-to-point Euclidean distance between the reconstructed dense 3D mesh and the real face for each of the users in the synthetic database, with and without the regularization term constraining the deformation. The reconstructions and the real faces can also be visually compared in Fig. 3.15a (no regularization) and Fig. 3.15b (regularization). Note that, since no texture information is available, all the faces are represented with the mean texture of the PCA.

Regarding the 3D error quantification, it is observed that better results are obtained when the reconstruction is constrained through a regularization term in the optimization, leading to a 3.62mm average error against the 4.47mm average error when no regularization is applied. There is some difference among users as well: users 1 and 8 have the worst reconstruction error, whereas users 2, 5, 6 and 7 achieve the best reconstructions.

If we look at the figures, we can visually observe the differences among users, although it is not easy to match the visual impression with the quantitative errors obtained in the table. In general, we can observe that regularized reconstructions result in faces closer to the mean face and probably more

Table 3.5: Full-head reconstruction errors for regularized and non-regularized optimization processes.

User	Full-Head Reconstruction Error (mm)	
	Non-regularized	Regularized
1	7,48	6,26
2	2,77	2,70
3	4,50	3,59
4	4,54	3,25
5	3,21	2,24
6	2,67	2,58
7	3,35	2,82
8	7,39	5,18
9	3,58	3,39
10	5,15	4,14
Avg	**4,47 ± 2.07**	**3,62 ± 1.66**

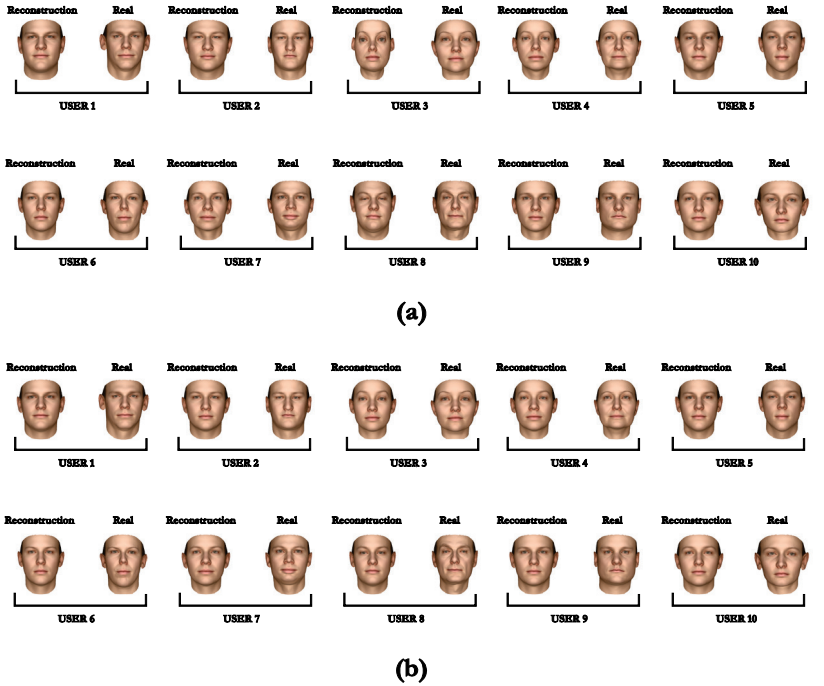

Figure 3.15: Comparison of full-head reconstructions and real faces. (a) Non-regularized optimization. (b) Regularized optimization.

homogeneous, whereas non-regularized reconstructions lead to more heterogeneous faces, sometimes possibly visually more recognizable because of the over-deformations, with the inherent risk of resulting in degenerate solutions (user 3 is an example of this).

All in all, it seems that 12 points are too few to give rise to accurate full-head reconstructions, since a lot of facial information is missing. However, it is interesting to see that, in some cases, even with only 12 points a reasonably recognizable full-head reconstruction can be achieved.

3.5 Concluding Remarks

This chapter addresses the problem of 3D face reconstruction based on 2D images, which is a necessary step for HPE. Different model fitting approaches have been proposed and described, some of them based on a single view of the face and some others making use of multiple views in different poses. All of them have been evaluated on the same framework, specifically on the synthetic database created on a completely controlled environment and presented in the previous chapter. This allows us to control all the variables affecting the face reconstruction, and provides us with the real 3D faces generated together with the database, which gives us a reference for fitting evaluation. All the methods have been evaluated and compared over two metrics: the 3D fitting error itself, measured as point-to-point Euclidean distance to the real face, and the HPE error obtained when using the reconstructed 3D shape with the POSIT algorithm. All the approaches have been evaluated both in noise-free 2D tracking conditions and in noisy tracking conditions, for different noise levels.

Regarding the evaluation results, several conclusions can be drawn:

- The global scale of the face significantly affects the fitting error, and this is something that cannot be corrected based only on 2D observations.

- Multi-view based reconstructions perform better than single-view reconstructions in terms of accuracy, although it is at the expense of computational cost.

- Accurate estimation of relative depths in the 3D shape seems critical to achieve accurate rotation estimations in HPE.

- The global scaling indetermination affects mainly translation estimation in HPE.

- As the 2D tracking noise increases, its effect on the HPE error tends to prevail over the effect of the inaccuracies in the reconstructed 3D model.

- In the presence of noisy 2D tracking, it is of critical importance for the HPE performance to obtain an alignment between the estimation and the ground truth. This can be achieved in two ways: 1) by registering the reconstructed shape to the real face (not possible in real situations); 2) by calculating a transformation for HPE based on a noise-free image frame (which can be achieved with a good initial detection of facial points in real situations).

- BA is the fitting method that obtains the best results in all the tests carried out.

A method to obtain the reconstruction of the full face based on the 12-point model has also been developed with promising results, although 12 points seem too few to reliably model the whole face.

Bibliography of the Chapter

[1] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Face Recognition Using Active Appearance Models," in *Proc. of the 5th European Conference on Computer Vision (ECCV '98)*, 1998, pp. 581–595.

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

[3] X. Zhang and Y. Gao, "Face Recognition Across Pose: A Review," *Pattern Recognit.*, vol. 42, no. 11, pp. 2876–2896, 2009.

[4] U. Park, Y. Tong, and A. K. Jain, "Age-Invariant Face Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 947–954, 2010.

[5] S. Ramanathan, A. Kassim, Y. V. Venkatesh, and W. S. Wah, "Human Facial Expression Recognition using a 3D Morphable Model," in *IEEE International Conference on Image Processing*, 2006, pp. 661–664.

[6] V. Blanz, C. Basso, T. Poggio, and T. Vetter, "Reanimating Faces in Images and Video," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 641–650, 2003.

[7] A. Maejima, S. Wemler, T. Machida, M. Takebayashi, and S. Morishima, "Instant Casting Movie Theater: the Future Cast System," *IEICE Trans. Inf. Syst.*, pp. 1135–1148, 2008.

[8] S. Baker, I. Matthews, and J. Schneider, "Automatic Construction of Active Appearance Models as an Image Coding Problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1380–1384, 2004.

[9] B. Gökberk, A. A. Salah, N. Alyüz, and L. Akarun, "3D Face Recognition: Technology and Applications," in *Handbook of Remote Biometrics, Advances in Pattern Recognition, Springer*, 2009, pp. 217–246.

[10] B. K. P. Horn, "Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View," 1970.

[11] J. J. Atick, P. a Griffin, and a N. Redlich, "Statistical Approach to Shape from Shading: Reconstruction of Three-Dimensional Face Surfaces from Single Two-Dimensional Images.," *Neural Comput.*, vol. 8, no. 6, pp. 1321–1340, 1996.

[12] R. Z. R. Zhang, P.-S. T. P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape from Shading: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 690–706, 1999.

[13] W. a P. Smith and E. R. Hancock, "Recovering Facial Shape Using a Statistical Model of Surface Normal Direction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1914–1929, 2006.

[14] A. Thelen, S. Frey, S. Hirsch, and P. Hering, "Improvements in Shape-From-Focus for Holographic Reconstructions With Regard to Focus Operators, Neighborhood-Size, and Height Value Interpolation," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 151–157, 2009.

[15] M. Castelán and E. R. Hancock, "Acquiring Height Data from a Single Image of a Face Using Local Shape Indicators," *Comput. Vis. Image Underst.*, vol. 103, no. 1, pp. 64–79, 2006.

[16] M. Castelán, W. a P. Smith, S. Member, E. R. Hancock, and a F. Shape-from-shading, "A Coupled Statistical Model for Face Shape Recovery From Brightness Images," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1139–1151, 2007.

[17] M. Black and A. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *Int. J. Comput. Vis.*, vol. 26, no. 1, pp. 63–84, 1998.

[18] F. D. La Torre, J. Vitria, P. Radeva, and J. Melenchon, "Eigenfiltering for Flexible Eigentracking (EFE)," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 2000, vol. 3, pp. 3–6.

[19] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models-Their Training and Application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[20] T. F. Cootes and C. J. Taylor, "Active Shape Models - 'Smart Snakes,'" in *Proc. British Machine Vision Conference (BMVC)*, 1992, pp. 266–275.

[21] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," in *Proc. European Conference on Computer Vision (ICCV)*, 1998, vol. 2, pp. 484–498.

[22] T. Cootes, G. Wheeler, K. Walker, and C. Taylor, "Coupled-View Active Appearance Models.," in *British Machine Vision Conference*, 2000, vol. 1, pp. 52–61.

[23] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[24] I. Matthews and S. Baker, "Active Appearance Models Revisited," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 135–164, 2004.

[25] F. De Torre and M. H. Nguyen, "Parameterized Kernel Principal Component Analysis : Theory and Applications to Supervised and Unsupervised Image Alignment," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, 2008.

[26] G. Tzimiropoulos, J. Alabort-I-Medina, S. Zafeiriou, and M. Pantic, "Generic Active Appearance Models Revisited," in *Proc. Assian Conference on Computer Vision (ACCV)*, 2013, vol. 7726 LNCS, pp. 650–663.

[27] V. Blanz and T. Vetter, "A Morphable Model For The Synthesis Of 3D Faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, 1999, pp. 187–194.

[28] L. Gu and T. Kanade, "3D Alignment of Face in a Single Image," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, vol. 1, pp. 1305–1312.

[29] V. Blanz, P. Grother, P. J. Phillips, and T. Vetter, "Face Recognition Based on Frontal Views Generated from Non-Frontal Images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 454–461.

[30] T. F. Cootes, K. Walker, and C. J. Taylor, "View-based active appearance models," *Proc. Fourth IEEE Int. Conf. Autom. Face Gesture Recognit. Cat No PR00580*, vol. 20, no. 9–10, pp. 657–664, 2000.

[31] A. K. Roy Chowdhury and R. Chellappa, "Face Reconstruction from Monocular Video using Uncertainty Analysis and a Generic Model," *Comput. Vis. Image Underst.*, vol. 91, pp. 188–213, 2003.

[32] D. Fidaleo and G. Medioni, "Model-Assisted 3D Face Reconstruction from Video," *Anal. Model. Faces Gestures, Lect. Notes Comput. Sci.*, vol. 4778, pp. 124–138, 2007.

[33] Y. S. Y. Shan, Z. L. Z. Liu, and Z. Z. Z. Zhang, "Model-Based Bundle Adjustment with Application to Face Modeling," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2001, vol. 2, pp. 644–651.

[34] D. Jiang, Y. Hu, S. Yan, L. Zhang, H. Zhang, and W. Gao, "Efficient 3D Reconstruction for Face Recognition," *Pattern Recognit.*, vol. 38, pp. 787–798, 2005.

[35] Z. Zhang, Z. Liu, D. Adler, M. F. Cohen, E. Hanson, and Y. Shan, "Robust and Rapid Generation of Animated Faces From Video Images: A Model-Based Modeling Approach," *Int. J. Comput. Vis.*, vol. 58, no. June, pp. 93–119, 2004.

[36] N. Faggian, A. P. Paplinski, and J. Sherrah, "Active Appearance Models for Automatic Fitting of 3D Morphable Models," in *Proc. IEEE International Conference on Video and Signal Based Surveillance (AVSS*, 2006.

[37] J. Heo and M. Savvides, "In Between 3D Active Appearance Models and 3D Morphable Models," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009, pp. 20–26.

[38] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-Time Combined 2D+3D Active Appearance Models," *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004.*, vol. 2, 2004.

[39] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: a Factorization Method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, 1992.

[40] M. Maruyama and S. Kurumi, "Bidirectional Optimization for Reconstructing 3D Shape from an Image Sequence with Missing Data," in *Proc. IEEE Int'l Conf. Image Processing*, 1999, pp. 120–124.

[41] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman, "The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences," *Int. J. Comput. Vis.*, vol. 32, no. 1, pp. 27–44, 1999.

[42] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering Non-Rigid 3D Shape from Image Streams," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, vol. 2, pp. 3–9.

[43] W. Brand, "Morphable 3D Models from Video," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, vol. 2, pp. 456–463.

[44] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler, "Tracking and Modeling Non-Rigid Objects with Rank Constraints," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 493–500.

[45] R. F. C. Guerreiro and P. M. Q. Aguiar, "3D Structure from Video Streams with Partially Overlapping Images," in *Proc. IEEE Int'l Conf. Image Processing*, 2002, pp. 897–900.

[46] L. Torresani, A. Hertzmann, and C. Bregler, "Learning Non-Rigid 3D Shape from 2D Motion," in *Proc. Neural Information Processing Systems (NIPS)*, 2003.

[47] A. Buchanan, "Investigation into Matrix Factorization when Elements are Unknown," 2004.

[48] A. Del Bue, F. Smeraldi, and L. Agapito, "Non-Rigid Structure from Motion using non-Parametric Tracking and Non-Linear Optimization," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[49] A. M. Buchanan and A. W. Fitzgibbon, "Damped Newton Algorithms for Matrix Factorization with Missing Data," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 316–322.

[50] M. Brand, "A Direct Method for 3D Factorization of Nonrigid Motion Observed in 2D," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 122–128.

[51] J. Xiao, J. Chai, and T. Kanade, "A Closed-Form Solution to Non-Rigid Shape and Motion Recovery," *Int. J. Comput. Vis.*, vol. 67, no. 2, pp. 233–246, 2006.

[52] C. Julia, A. Sappa, F. Lumbreras, J. Serrat, and L. Antonio, "Factorization with Missing and Noisy Data," in *International Conference on Computational Science*, 2006, pp. 555–562.

[53] S. I. Olsen and a. E. Bartoli, "Using Priors for Improving Generalization in Non-Rigid Structure-from-Motion," in *Proc of the British Machine Vision Conference*, 2007, pp. 105.1–105.10.

[54] D. Nistér, F. Kahl, and H. Stewénius, "Structure from Motion with Missing Data is NP-Hard," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–7.

[55] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid Structure-from-Motion: Estimating Shape and Motion with Hierarchical Priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 878–892, 2008.

[56] A. Del Bue, "A Factorization Approach to Structure from Motion with Shape Priors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[57] H. S. Koo and K. M. Lam, "Recovering the 3D Shape and Poses of Face Images Based on the Similarity Transform," *Pattern Recognit. Lett.*, vol. 29, no. 6, pp. 712–723, 2008.

[58] J. Fortuna and A. M. Martinez, "Rigid Structure from Motion from a Blind Source Separation Perspective," *Int. J. Comput. Vis.*, vol. 88, no. 3, pp. 404–424, 2010.

[59] J. Gonzalez-Mora, F. De La Torre, N. Guil, and E. L. Zapata, "Learning a Generic 3D Face Model from 2D Image Databases using Incremental Structure from Motion," *Image Vis. Comput.*, vol. 28, no. 7, pp. 1117–1129, 2010.

[60] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Trajectory Space: A Dual Representation for Nonrigid Structure from Motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1442–1456, 2011.

[61] S. J. Lee, K. R. Park, and J. Kim, "A SfM-based 3D Face Reconstruction Method Robust to Self-Occlusion by using a Shape Conversion Matrix," *Pattern Recognit.*, vol. 44, no. 7, pp. 1470–1486, 2011.

[62] Z.-L. Sun, K.-M. Lam, and Q.-W. Gao, "Depth Estimation of Face Images Using the Nonlinear Least-Squares Model," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 17–30, 2013.

[63] J. Heo, "3D Generic Elastic Models for 2D Pose Synthesis and Face Recognition," PhD dissertation, Carnegie Mellon Univ., 2009.

[64] U. Prabhu, J. Heo, and M. Savvides, "Unconstrained Pose Invariant Face Recognition Using 3D Generic Elastic Models.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1952–1961, 2011.

[65] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, 2000.

[66] J. Xiao, T. Moriyama, T. Kanade, and J. F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *Proc. Fifth IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2002.

[67] T. Kanade, J. Sung, and D. Kim, "Pose Robust Face Tracking by Combining Active Appearance Models and Cylinder Head Models," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 260–274, Jan. 2008.

[68] R. Valenti, N. Sebe, and T. Gevers, "Combining Head Pose and Eye Location Information for Gaze Estimation," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 802–815, 2012.

[69] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D Face Model for Pose and Illumination Invariant Face Recognition," *2009 Sixth IEEE Int. Conf. Adv. Video Signal Based Surveill.*, pp. 296–301, Sep. 2009.

[70] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions," *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–117, 1998.

[71] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle Adjustment - A Modern Synthesis," *Vis. Algorithms Theory Pract.*, vol. 1883, pp. 298–372, 2000.

[72] V. Rabaud, "Vincent's Structure from Motion Toolbox." [Online]. Available: http://github.com/vrabaud/sfm_toolbox.

CHAPTER 4

Pose Estimation

This chapter focuses on the topic of 3D pose estimation based on 2D point tracking and proposes several contributions to enhance the performance based on using combined information from 2D and 3D. It is addressed as a problem of generic object pose estimation, in order to show that the methods and improvements proposed in this work are valid not only for HPE but for any object for which an approximate 3D model is available. We believe this is an important added value of this thesis, even though the original main goal has been to contribute to HPE methods. Section 4.1 presents the basic approach for estimating the pose of an object using the POSIT algorithm. Section 4.2 proposes and validates the novel 'weighted POSIT' (wPOSIT) algorithm, a modification of the original POSIT algorithm that aims to enhance its performance by applying weights to the 2D-3D point correspondences based on the tracking accuracy. Section 4.3 develops a method to obtain a tracking accuracy index (TAI) that can be used for weight calculation in the wPOSIT algorithm, based on two invariant shape metrics that we propose. Section 4.4 addresses the problem of outlier detection and correction in 2D images using combined information from 2D and 3D. Finally, Section 4.5 is focused on applying the presented methods to the specific problem of HPE, describing the adaptations proposed and evaluating 2D tracking and 3D HPE results on two different head pose databases of videos. Closing the chapter, Section 4.6 presents the concluding remarks.

4.1 Basic Approach

This chapter is going to present various methods and improvements oriented to obtaining more accurate 2D tracking and 3D pose estimation, all of them based on using the POSIT algorithm [1] already presented in this thesis. We will thus call the "basic approach" to applying POSIT, using a set of 2D-3D correspondent points that describe the shape of the object, in order to obtain its 3D pose in a sequence of images. We will define $\boldsymbol{x}_i^p = \left(x_i^p, y_i^p\right)$ the set of 2D points in the sequence, with $i = 1 \dots N$ frames, and $\boldsymbol{X}^p = (X^p, Y^p, Z^p)$ the set of corresponding 3D points in the model, with $p = 1 \dots P$ the number of point correspondences that describe the shape.

The set of 3D points will be obtained through a 3D shape reconstruction algorithm based on the 2D observations from different frames, what is called in the literature structure from motion (SfM). SfM includes methods such as the ones described in Chapter 3. While the 3D point set is fixed for the object once the reconstruction has been performed, the corresponding 2D points must be obtained for each video frame for which the pose of the object is going to be estimated. This is usually achieved by an initial 2D feature point detection, typically using a frontal view of the object, followed by a 2D tracking that updates the location of the detected features in the image according to the object motion. The tracked 2D points and the previously reconstructed 3D points are therefore passed to the POSIT algorithm in order to determine the 3D rotation and translation that define the pose of the object in every frame of an image sequence, i.e. $(\boldsymbol{R}_i, \boldsymbol{T}_i)$ with $i = 1 \dots N$.

One of the main conclusions drawn from the previous chapter has been that inaccurate 2D tracking leads to 3D pose estimation errors, and that 2D feature location errors actually prevail over 3D model inaccuracies in the resultant pose, showing a more significant contribution to the error. It is thus of critical importance to find solutions that either correct the 2D tracking or at least compensate to the possible extent the effect that inaccurate image feature location causes in the 3D pose estimation. These are the main issues that are going to be addressed in the next sections and for which different solutions are going to be proposed.

4.2 Weighted POSIT

This section presents a novel variant of the POSIT algorithm, what we call the 'weighted POSIT' (wPOSIT). The idea is to improve the performance of the algorithm through the introduction of weights applied to the point correspondences. The proposed method will be evaluated using a simulated environment that has been built for this purpose.

4.2.1 Algorithm Description

The wPOSIT is a modified version of the POSIT algorithm, in which a certain weight is applied to each 2D-3D point correspondence. The manner in which these weights are applied consists in repeating each point as many times as its weight indicates. That is, the list of 2D-3D point correspondences passed to the POSIT algorithm contains repeated points according to the weights assigned. Therefore, the weights are integer numbers and the maximum weight is a parameter of the algorithm that, in turn, determines the number of weight levels available.

The optimal way to determine the weight of a point in a certain frame would be to calculate a tracking accuracy index (TAI), which should indicate the accuracy with which the point has been located in that frame; the more accurately the point has been tracked, the higher the weight that should be applied, and vice versa. Obtaining a TAI for each point in every frame is a challenging task that will be addressed in Section 4.3. For the moment, we will assume that this index has already been obtained and we will apply ideal weights to the points in order to assess the performance of the wPOSIT against the basic approach (i.e. the non-weighted version of POSIT).

4.2.2 Simulation Results

The evaluation of the wPOSIT has been carried out with the aid of the simulator tool described in Chapter 2. Therefore, the simulation to assess the validity of wPOSIT has been performed in the specific environment of HPE. Similarly to what has been done in Section 3.3.1 of Chapter 3, Gaussian noise of mean zero and selected values of standard deviation has been applied to the 2D ground truth of the synthetic database presented in Chapter 2 to generate a set of noisy databases, simulating noisy tracking. The PDF of this noise is described by equation (3.15) from Chapter 3. Using equation (3.18), the standard deviation values have been selected so that the average tracking error, measured as

Euclidean distance from the noisy 2D observations to the ideal ones, takes round values ranging from 0 to 8 pixels for the different noisy databases generated. The image resolution of these synthetic databases is 1280×720 pixels. The 0-8 pixel range has been selected based on preliminary experimentation, which has shown us that errors between 3-7 pixels seem typical for that resolution when using some of the most standard 2D tracking methods, such as Lucas-Kanade [2], IntraFace [3], ASM [4] or AAM [5].

Using the ideal 3D model of each user in the synthetic database and the corresponding 2D projections in each frame, POSIT and wPOSIT have been applied to all the noisy databases generated, including the noise-free database, in order to obtain HPE errors for comparison. The weight of each point in each frame has been calculated using a linear function that relates it to the distance from the noisy point to its ideal location in the image, assigning the lowest weight (i.e. 1) to the maximum observed distance in the database, and the highest weight when the distance is zero. It has been observed in preliminary experiments that the number of weight levels has an almost negligible impact on the pose estimation accuracy for a wide range of levels (10-100). Hence, the number of weight levels has been set to 50 in this experiment.

The comparison between POSIT and wPOSIT is shown in Fig. 4.1. As it could be expected, the average HPE error (translation and rotation) of both approaches is identical for the noise-free database, since the 2D points are ideal and thus the same weight is applied to any point in the database, which makes wPOSIT perform as a non-weighted POSIT. This error is also identical to the one presented in Chapter 3 for ideal 2D-3D correspondences, i.e. 0.35mm in translation and 0.04° in rotation. It is also observed that, as the tracking noise increases, HPE errors for both methods increase, and so does the absolute difference between them. In fact, if we measure the enhancement provided by wPOSIT with respect to POSIT as a ratio, we obtain an approximately stable 10% HPE accuracy improvement for any tracking noise level.

We can conclude from the presented experiment that, if we are able to ideally estimate how accurately a point is being tracked in a sequence, we can improve our pose estimation accuracy by a 10% if we apply wPOSIT instead of the non-weighted POSIT. Therefore, it has been shown that wPOSIT is a valid and very interesting contribution to pose estimation methods.

However, there is an important consideration that should be made regarding this experiment. It is important to note that this simulation has

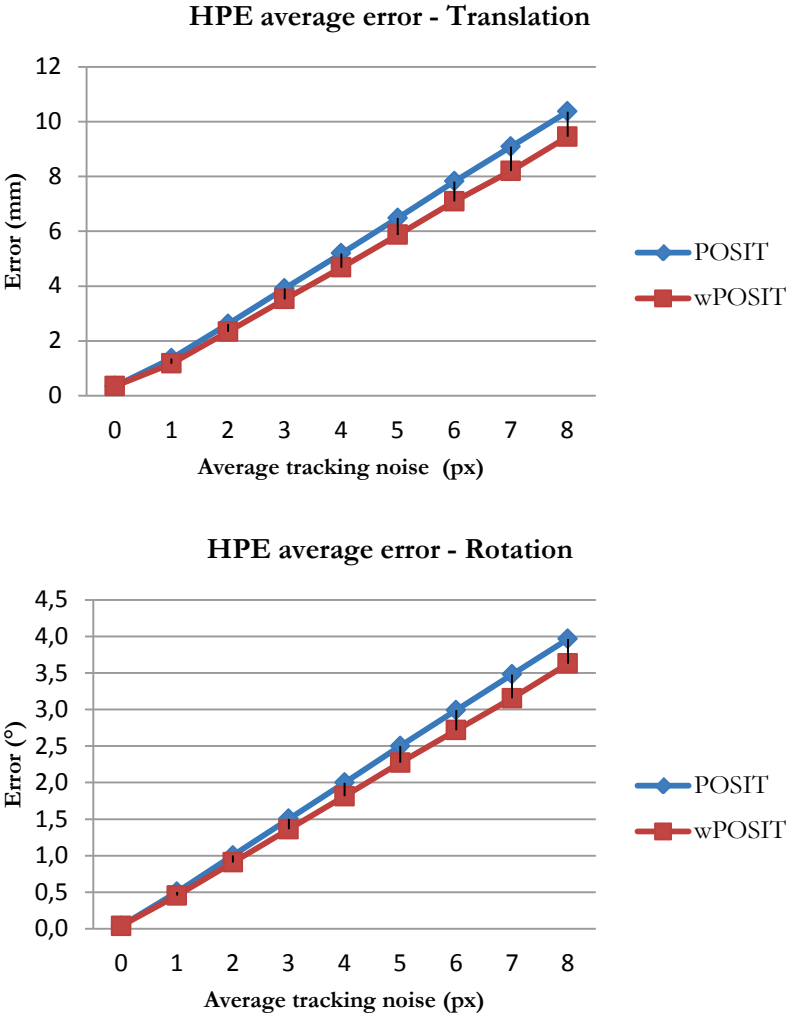

Figure 4.1: POSIT and wPOSIT performance comparison. Average HPE error (rotation and translation) for different 2D tracking noise levels is shown.

approximated the 2D tracking noise as a Gaussian noise affecting all the points equally. This means that there may be several frames where many points in the shape are inaccurately tracked and, although weights are distributed in the best possible way to compensate for the effect of those tracking errors in the pose

estimation, most of the points may receive a middle-range weight and the performance of wPOSIT may not differ much from POSIT for those frames. Nevertheless, tracking error in real situations may not be accurately described by a Gaussian distribution. When all the points are being accurately tracked except one or two that have hooked up to a static point in the image, due for instance to a partial occlusion of that part of the shape, the improvement provided by the wPOSIT may be magnified. HPE given by POSIT would suffer considerably in that situation, since a few points would present a very high tracking noise, but wPOSIT would be able to maximize its performance by applying a very high weight to most of the points and a very low weight to the few ones that have lost track.

4.3 Tracking Accuracy Index (TAI)

This section presents a method to obtain a TAI based on the use of two invariant shape metrics and their associated tolerance models. In order to ensure the feasibility of the method, object shape and pose independency must be achieved. Hence, methods to overcome these problems are also proposed. Finally, the weight calculation using the TAI is described for wPOSIT.

4.3.1 Invariant Shape Metrics

The proposed method is based on using interlandmark relationships to analyze the 2D geometrical configuration of the tracked points for each videoframe. This idea was introduced by Lekadir et al. in [6], where the ratio of interlandmark distances in triplets of points was defined as an invariant shape metric. In this thesis, we propose two complementary shape metrics (*r* and *s*) to characterize the geometrical configuration of the 2D features tracked in the object. These metrics are defined for each triplet of 2D points (p^j, p^k, p^l) as

$$r^{jkl} = \frac{d^{jk}}{d^{kl}} \quad , \quad s^{jkl} = \theta(\overrightarrow{jk}, \overrightarrow{kl}) \qquad (4.1)$$

where d^{jk} and d^{kl} represent the Euclidean distance from p^j to p^k and from p^k to p^l respectively, and $\theta(\overrightarrow{jk}, \overrightarrow{kl})$ is the angle formed by two vectors, one going from p^j to p^k and the other one from p^k to p^l. In fact, six shape metrics are defined for each triplet of points, i.e. $(r^{jkl}, r^{klj}, r^{ljk})$ and $(s^{jkl}, s^{klj}, s^{ljk})$. The

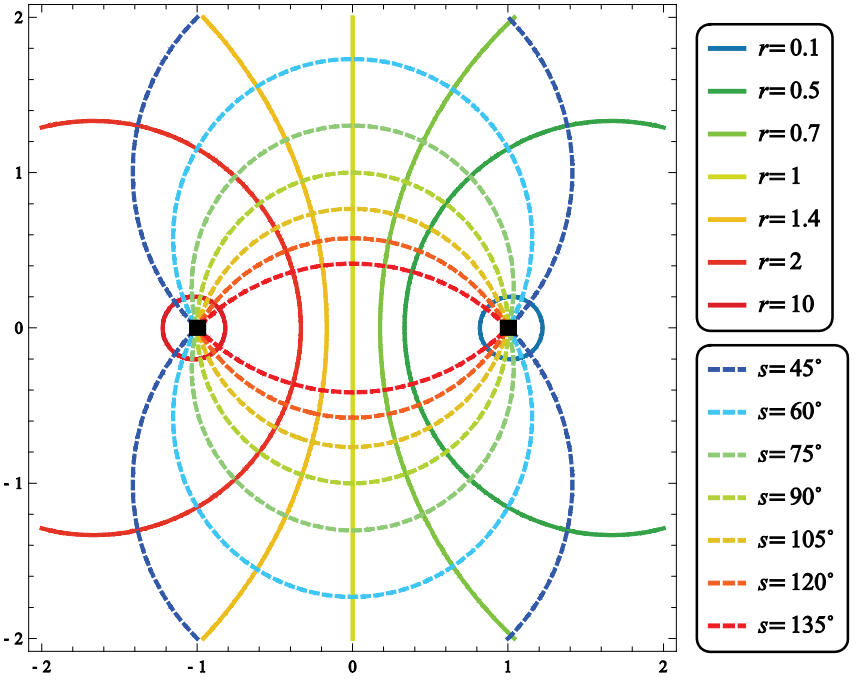

Figure 4.2: Complementarity of the shape metrics *r* and *s*. Two fixed points, (-1,0) and (1,0) are represented, and around them a set of contours where the third point of the triplet may lie for each of the metrics to keep a constant value.

advantage of using these two shape metrics is that they are complementary, in the sense that any 2D point that drifts from its original position in the image produces a change in at least one of the two metrics.

Fig. 4.2 shows this complementarity: fixing two points of a certain triplet, (-1,0) and (1,0) in the figure, for each value of r there is a contour along which the third point of the triplet may drift while keeping r constant. This means that there is a possibility of a point drifting along a specific contour in the object during the tracking so that this drifting would go undetected for the first shape metric. An analogous behavior is observed for the metric s, as shown also in Fig. 4.2. However, the contours that keep each metric constant are different and intersect only in two specific points in the image. Therefore, there does not exist any contour along which a point may drift and go undetected for both shape metrics, and it is very unlikely that the tracking error will consist in a point

jumping from one intersection of the corresponding contours to the other intersection, which would be the only possibility where the two shape metrics would not detect a tracking inaccuracy.

These metrics thus describe the relative geometrical configuration of the 2D points detected in the object, and therefore each point can be analyzed from its relative position with respect to other points in the shape. This information is very useful because it can be managed with the aim of distinguishing the points that are inaccurately located in the shape from those that have been tracked correctly. However, getting to that point is not straightforward, since each metric is making reference to three image points and there is no way of saying which one or which ones are incorrect just from the metric value. Moreover, these metrics show the advantage of being invariant to 2D scaling, rotation or translation, but they are not invariant to 3D pose changes in the object and neither to differences in appearance of objects of the same kind. The latter is easier to explain with an example: if we are working with facial features, and we define a triplet consisting for instance of two eye corners and the nose tip, both shape metrics may have different values for different persons even if the tracking is perfect, just due to differences in their facial appearance. This is true even if we are approximating the two persons with one generic 3D face model. Same reasoning is applicable for any object we aim to track, such as cars of different models, planes, buildings, etc.

The following sections describe the solutions proposed to overcome these issues and achieve a full invariability of the shape metrics, which is one of the main challenges of the proposed method.

4.3.2 Pose Normalization

In order to make the shape metrics invariant with respect to the 3D pose of the object, thus minimizing the 2D appearance difference due to pose changes, the tracked 2D points are pose-normalized to a reference pose, usually a frontal view of the object. The process of pose normalization can be observed in Fig. 4.3, illustrated for the specific example of face tracking and head pose normalization. It consists in simulating a camera rotation and translation in order to get a frontal view of the face. First, the 3D pose of the object is estimated using POSIT and all the tracked points that define the object shape. In the example in the figure, the estimated pose consists in the transformation between the head (X_M, Y_M, Z_M) and the camera (X_c, Y_c, Z_c) coordinate systems, being the camera in its actual position (before the pose-normalization). Using that information, each of the 2D image points is back-projected to the 3D model in

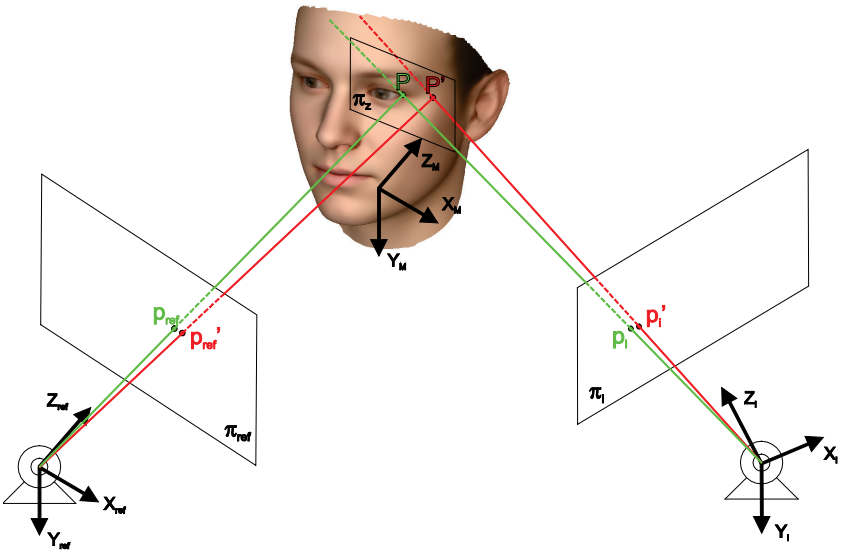

Figure 4.3: Illustration of the pose normalization process. Original camera (X_i,Y_i,Z_i) and pose-normalized camera (X_{ref},Y_{ref},Z_{ref}) are represented, together with the image projections of a correctly-tracked point P and a wrongly-tracked point P' from the 3D head model coordinate system (X_M,Y_M,Z_M).

the calculated pose, and the intersection between the back-projection line and the *Z*-plane corresponding to the depth of the original model point (π_Z plane in Fig. 4.3) is obtained as the current 3D object point being tracked in the image. Finally, this new 3D point is reprojected to a camera located just in front of the object according to the calculated pose (image plane π_{ref} in Fig. 4.3). In the example in the figure, an inaccurately tracked (p_i') and ideally tracked (p_i) corner of the eye is shown, thus two image representations (one correct and one incorrect) of the same anatomical point. For the inaccurate image point, the back-projection line is drawn and the intersection with the π_Z plane is shown (P'). The π_Z plane corresponds to the depth of the ideal model point P in the head coordinate system (X_M,Y_M,Z_M), which is the 3D model point corresponding to the ideally tracked point p_i. Both image points in the original camera plane, i.e. p_i and p_i' in the π_i camera plane, give place to two different image points in the calculated frontal view, i.e. p_{ref} and p_{ref}' in the π_{ref} camera plane. The distance in the new *Z* axis (Z_{ref} in Fig. 4.3) at which the frontal camera is located is irrelevant since both shape metrics are scale invariant in 2D. The new image points represent the pose-normalized appearance of the tracked features. Note in the example in the figure

that the pose-normalized reprojections of the ideally tracked point (p_{ref}) and the inaccurately tracked point (p_{ref}') keep being representative of the tracking error committed, but the difference between both points will only be due to the 2D tracking error and not to the instantaneous 3D pose of the head, if we assume an ideal pose estimation. In reality, this is not possible and the error in the 3D pose estimation will inevitably affect to some length the pose normalization process and the resulting 2D points, making impossible to achieve a full pose independency. Nonetheless, it will be later shown in the results section that this error is acceptable and overall the method performs well, overcoming the 3D pose dependency problem.

4.3.3 Object Independency

Object independency refers to the fact that the shape metrics should not depend on the specific appearance of the current object, which may not be identical to another object of the same kind. This is also easily understood with the example of faces, where metrics described by the same anatomical points will differ from one person to another just due to differences in their facial configuration (eye separation, mouth width…). In order to overcome this problem, the initial frame of the sequence to track is chosen as a reference template. The assumption made here is that our initial 2D feature point detection is correct and can therefore be used as a reference for comparison. The initial configuration of the points is then taken as 'ideal' for that object, and *r* and *s* metric values in every frame can be compared with that reference template, using the following comparison metrics:

$$cr_i = \begin{cases} \dfrac{r_i}{r_{ref}} & if\ r_i \geq r_{ref} \\ -\dfrac{r_{ref}}{r_i} & if\ r_i < r_{ref} \end{cases} \quad , \qquad cs_i = s_i - s_{ref} \qquad (4.2)$$

being r_i and s_i the shape metrics calculated for the i^{th} frame as defined in equation (4.1), and r_{ref} and s_{ref} the same metrics calculated for the reference frame (1st frame). The superscript that refers to the point number shown in equation (4.1) is not shown in equation (4.2) for simplicity in notation. The comparison metric cr_i is defined in two sections in order to get two symmetric ranges for the values of the metric, i.e. (-∞,-1) and [1, ∞).

The metrics for every frame in the sequence, r_i and s_i, are obtained using the previously pose-normalized images. That is, both metrics are calculated

using the frontal view of the tracked points so that 2D differences due to 3D pose variations are avoided, as described in the previous section. This is also done with the metrics for the first frame, r_{ref} and s_{ref}. Working with the comparison metrics defined in equation (4.2) instead of the absolute metrics defined in equation (4.1) allows us to compare the interlandmark configuration of the current frame with the interlandmark configuration of the initial frame that is taken as reference. This comparison can be carried out whatever the 3D pose difference between both frames is, since the 2D shape points have previously been pose-normalized. And measuring the relative differences between both interlandmark configurations (as is done in equation (4.2)) assures that appearance differences between different objects of the same kind are not affecting the measure. Therefore, we can claim that, through the described process, we obtain two invariant shape metrics (i.e. cr_i and cs_i) that only depend on the accuracy with which the feature points in the object are being tracked. Note that, in perfect tracking conditions, $cr_i = 1$ and $cs_i = 0$ for any triplet of points.

4.3.4 Tolerance Model

Each landmark is now associated with two sets of invariant shape metrics for each frame. The main idea behind the calculation of a TAI is that an inaccurately tracked point will cause metrics calculated from triplets that include that point to be invalid. In order to tell whether a certain metric value is accepted or not, tolerance intervals (T) that characterize a valid model have to be calculated in what is usually called tolerance analysis [7]. Extreme values of the metrics can then be detected if they fall outside the statistically determined tolerance intervals, usually calculated from training samples.

If we denote F the cumulative distribution function of a metric c, being c the generic notation for the metrics cr and cs for a certain triplet of points, we can define its corresponding statistical tolerance interval $T = [c_L, c_H]$ as a two-sided interval with β-content and γ-confidence if

$$P[F(c_H) - F(c_L) \geq \beta] = \gamma \qquad (4.3)$$

The meaning of this equation is that at least a proportion β of the total population lies in the interval $[c_L, c_H]$ with a confidence coefficient equal to γ. According to [8], if we assume that the metric c follows a Gaussian distribution, the statistical tolerance interval that solves equation (4.3) can be defined as

$$T = [\mu_c - k_2\sigma_c, \mu_c + k_2\sigma_c] \tag{4.4}$$

where μ_c and σ_c are the mean and standard deviation of the distribution respectively, and k_2 is the two-sided tolerance factor that can be approximated using the following equation according to [9]

$$k_2 = z_{(1-\beta)/2}\sqrt{\left(1+\frac{1}{N}\right)\frac{N-1}{\chi^2_{\gamma,N-1}}} \tag{4.5}$$

where N is the sample size, $z_{(1-\beta)/2}$ the upper $(1-\beta)/2$ quantile of the standard normal distribution and $\chi^2_{\gamma,N-1}$ the lower γ quantile of the chi-squared distribution with $N-1$ degrees of freedom.

Even though a Gaussian distribution of the samples is known, or often assumed, in many applications, there are cases where this could be a bad approach and it is better to use non-parametric tolerance intervals. According to [10], these should be calculated from the smallest and largest observation in the training set

$$T = [min(c), max(c)] \tag{4.6}$$

Note that each triplet of points in the shape (j,k,l) gives room to two different shape metrics (cr^{jkl} and cs^{jkl}), and these will in turn have their own tolerance intervals associated (T_{cr}^{jkl} and T_{cs}^{jkl}).

4.3.5 Weight Calculation

Once the tolerance model has been built, two steps are followed in order to measure the dissimilarity of a point with the model. First, a likelihood measure f_c is obtained for each metric. Typically, this measure is defined as binary, being 1 when the metric lies inside the tolerance interval and 0 otherwise

$$f_c(c^{jkl}) = \begin{cases} 1, & if\ c^{jkl} \in T^{jkl} \\ 0, & elsewhere \end{cases} \tag{4.7}$$

In our case, and according to how the comparison metrics have been defined in equation (4.2), we build the likelihood measure for each metric as a

linear function with values going from 1 to 0 inside the tolerance interval, being 1 when the metric corresponds to the perfect tracking, and 0 outside the interval:

$$f_{cr}(cr) = \begin{cases} \dfrac{cr_H - cr}{cr_H - 1}, & if\ 1 \le cr \le cr_H \\ \dfrac{cr_L - cr}{cr_L + 1}, & if\ -1 > cr \ge cr_L \\ 0, & elsewhere \end{cases} \tag{4.8}$$

$$f_{cs}(cs) = \begin{cases} 1 - \dfrac{cs}{cs_H}, & if\ 0 \le cs \le cs_H \\ 1 - \dfrac{cs}{cs_L}, & if\ 0 > cs \ge cs_L \\ 0, & elsewhere \end{cases} \tag{4.9}$$

where tolerance intervals corresponding to both metrics are defined for each triplet as $T_{cr} = [cr_L, cr_H]$ and $T_{cs} = [cs_L, cs_H]$. Note that the superscript 'jkl' corresponding to each triplet is not shown in the equations for clarity of notation.

When a metric falls outside of the tolerance, it is not straightforward to identify which point in the triplet is the cause, but we can calculate the likelihood f^p of each point to be an inlier by adding up the contributions of all the metrics associated to the point:

$$f^p(p^j) = \frac{1}{N^j} \sum_{k,l} f_c(c^{jkl}) f_c(c^{klj}) f_c(c^{ljk}) \tag{4.10}$$

where N^j is the number of triplets in which the point p^j is present. Here again we use a general notation for both metrics, but we will actually obtain two likelihood measures (fr^p and fs^p) applying equation (4.10) to the two metrics. The two likelihood measures are averaged to get the final f^p for every landmark.

This process is carried out for each frame in the video sequence, and hence the likelihood measure can be written as f_i^p, where $i = 1 \dots N$ frames and $p = 1 \dots P$ points defining the shape of the object. In fact, f_i^p is the TAI we were looking for from the beginning, ranging from 1, when the point is perfectly tracked, to 0, when all the metrics in which the point is involved fall outside the tolerance model.

Algorithm 4.1 wPOSIT

Input: $\boldsymbol{X}^p: (X^p, Y^p, Z^p)$, $\boldsymbol{x}_i^p: (x_i^p, y_i^p)$; $p = 1 \dots P$ points, $i = 1 \dots N$ frames. We denote them as $\boldsymbol{X}$ ($P \times 3$ matrix) and $\boldsymbol{x}_i$ ($P \times 2$ matrix) for simplicity.
Process initial frame (we denote it as *ref* for 'reference'):
    Apply POSIT: $(\boldsymbol{R}_{ref}, \boldsymbol{T}_{ref}) = POSIT(\boldsymbol{X}, \boldsymbol{x}_{ref})$
    Pose-normalized image points: $\tilde{\boldsymbol{x}}_{ref} = PoseNorm(\boldsymbol{R}_{ref}, \boldsymbol{T}_{ref}, \boldsymbol{x}_{ref})$
    Calculate r_{ref} and s_{ref} invariant shape metrics from $\tilde{\boldsymbol{x}}_{ref}$
for $i = 2 \rightarrow N$ **do** (process rest of frames)
    Apply POSIT: $(\boldsymbol{R}_i, \boldsymbol{T}_i) = POSIT(\boldsymbol{X}, \boldsymbol{x}_i)$
    Pose-normalized image points: $\tilde{\boldsymbol{x}}_i = PoseNorm(\boldsymbol{R}_i, \boldsymbol{T}_i, \boldsymbol{x}_i)$
    Calculate r_i and s_i invariant shape metrics from $\tilde{\boldsymbol{x}}_i$
    Obtain comparison metrics: $cr_i = f(r_{ref}, r_i)$ and $cs_i = f(s_{ref}, s_i)$
    Tolerance analysis to obtain TAI: $f_i = TolAnalysis(cr_i, cs_i)$
    Calculate weights: $w_i = WeightCalc(f_i)$
    Apply wPOSIT: $(\boldsymbol{R}_i', \boldsymbol{T}_i') = wPOSIT(\boldsymbol{X}, \boldsymbol{x}_i, w_i)$
end for
Output: $(\boldsymbol{R}_i', \boldsymbol{T}_i')$; $i = 1 \dots N$ frames

This TAI can easily be transformed into a point weight for wPOSIT by applying weights proportional to the TAI value, knowing that the minimum weight (i.e. 1) must correspond to a TAI of 0, and the maximum weight w_{max} (i.e. the number of weight levels we want to use) to a TAI of 1. Weights are thus obtained from TAIs through the following equation:

$$w^p = round(1 + (w_{max} - 1) \cdot f^p) \qquad (4.11)$$

Note that the result of the equation is rounded in order to obtain integer weights, because it must be reminded that weights are applied in wPOSIT by point repetition.

The complete wPOSIT method developed in this thesis, including the whole process for TAI calculation described above, is summarized in Algorithm 4.1.

4.4 Outlier Detection and Correction

While in the previous section we have explained how to obtain weights to apply wPOSIT, so that poorly tracked points have lower weight in the 3D pose estimation than accurately tracked ones, this section aims to go one step further and detect and eliminate outliers in the 2D shape tracked in each frame. Moreover, a method to correct these outliers is presented, aiming to improve the 2D tracking performance and hence the subsequent 3D pose estimation. Both outlier handling methods are combined with wPOSIT and are actually an extension of Algorithm 4.1.

4.4.1 Outlier Detection (OD)

Outlier detection (OD) can be carried out after the TAI calculation presented in the previous section. We need to set a threshold λ for the TAI below which a point will be considered a potential outlier. This condition would then be defined by the following equation

$$f^p(p^j) < \lambda \qquad (4.12)$$

The process of OD is described in detail in Algorithm 4.2. In fact, OD is combined with the wPOSIT method described in Algorithm 4.1, and lines 12-24 of Algorithm 4.2 correspond to the OD module in the whole wPOSIT+OD method. After TAI calculation for all the feature points in a certain frame, the point with the minimum TAI is compared with the threshold λ. If equation (4.12) is satisfied, the point is considered a potential outlier and undergoes the final checking process that will confirm or discard it as an actual outlier. This process can be divided in three steps: first, the TAIs are recalculated using only the metrics cr and cs of triplets that do not include the potential outlier, and new weights are calculated from these TAIs for the 2D point set in which the outlier is not included; then, wPOSIT is applied with the new weights to obtain a pose estimation that is supposed to be more accurate since the outlier has been excluded from the calculation, and this new pose estimation is used to repeat the steps of pose normalization, new metrics' calculation and final TAIs estimation (without the outlier once again); finally, these new TAIs are compared with the original ones (excluding the outlier) and, if the average value of the new TAIs exceeds the average value of the old ones, the outlier is confirmed. If this final condition is met, it means that the metrics in which the outlier was involved were contributing negatively to the TAIs of the other landmarks, which is a very

Algorithm 4.2 wPOSIT+OD

Input: $\boldsymbol{X}^p: (X^p, Y^p, Z^p)$, $\boldsymbol{x}_i^p: (x_i^p, y_i^p)$; $p = 1 \dots P$ points, $i = 1 \dots N$ frames. We denote them as $\boldsymbol{X}$ ($P \times 3$ matrix) and $\boldsymbol{x}_i$ ($P \times 2$ matrix) for simplicity.
Process initial frame (we denote it as *ref* for 'reference'):
    Apply POSIT: $(\boldsymbol{R}_{ref}, \boldsymbol{T}_{ref}) = POSIT(\boldsymbol{X}, \boldsymbol{x}_{ref})$
    Pose-normalized image points: $\tilde{\boldsymbol{x}}_{ref} = PoseNorm(\boldsymbol{R}_{ref}, \boldsymbol{T}_{ref}, \boldsymbol{x}_{ref})$
    Calculate r_{ref} and s_{ref} invariant shape metrics from $\tilde{\boldsymbol{x}}_{ref}$
for $i = 2 \rightarrow N$ **do** (process rest of frames)
    Apply POSIT: $(\boldsymbol{R}_i, \boldsymbol{T}_i) = POSIT(\boldsymbol{X}, \boldsymbol{x}_i)$
    Pose-normalized image points: $\tilde{\boldsymbol{x}}_i = PoseNorm(\boldsymbol{R}_i, \boldsymbol{T}_i, \boldsymbol{x}_i)$
    Calculate r_i and s_i invariant shape metrics from $\tilde{\boldsymbol{x}}_i$
    Obtain comparison metrics: $cr_i = f(r_{ref}, r_i)$ and $cs_i = f(s_{ref}, s_i)$
    Tolerance analysis to obtain TAI: $f_i = TolAnalysis(cr_i, cs_i)$
    while $(min(f_i) < \lambda)$ **do**
        Repeat steps 9-11 without outlier and obtain f_i^{OD}
        Calculate weights without outlier: $w_i^{OD} = WeightCalc(f_i^{OD})$
        Apply wPOSIT without outlier: $(\boldsymbol{R}_i^{OD}, \boldsymbol{T}_i^{OD}) = wPOSIT(\boldsymbol{X}^{OD}, \boldsymbol{x}_i^{OD}, w_i^{OD})$
        Pose-normalized image points: $\tilde{\boldsymbol{x}}_i^{OD} = PoseNorm(\boldsymbol{R}_i^{OD}, \boldsymbol{T}_i^{OD}, \boldsymbol{x}_i^{OD})$
        Repeat steps 9-11 with $\tilde{\boldsymbol{x}}_i^{OD}$ and obtain the final f_i^{OD}
        if $\left(mean(f_i^{OD}) > mean(f_i)\right)$ **then**
            Confirm and eliminate outlier. Update $f_i = f_i^{OD}$, $\boldsymbol{x}_i = \boldsymbol{x}_i^{OD}$
        else
            Break **while loop**
        end if
    end while
    Calculate final weights: $w_i = WeightCalc(f_i)$
    Apply wPOSIT: $(\boldsymbol{R}_i', \boldsymbol{T}_i') = wPOSIT(\boldsymbol{X}, \boldsymbol{x}_i, w_i)$
end for
Output: $(\boldsymbol{R}_i', \boldsymbol{T}_i')$; $i = 1 \dots N$ frames

reasonable indicative of a point being an outlier. This OD loop is iteratively repeated until no more outliers are detected, checking the lowest TAI among the new ones against the threshold at each iteration. Once the outlier-free 2D-3D correspondences have been determined and the final TAIs have been calculated, wPOSIT is applied in order to obtain the definitive pose for the current frame.

4.4.2 Outlier Correction (OC)

As a final contribution, we present a method for outlier correction (OC). Once outliers have been detected in a certain frame, an alternative to just removing them is to try to readjust them to their correct image position. For this to be done, there is usually a need for some kind of reference template on which we can rely to calculate the ideal image position of the detected outlier.

The process of OC is described in detail in Algorithm 4.3. The OC module is actually integrated in the wPOSIT+OD method described in Algorithm 4.2, to give place to the final wPOSIT+OD+OC method. Lines 16-24 in Algorithm 4.3 correspond to the OC module integrated in the complete algorithm. Once a potential outlier is detected, new TAIs are obtained excluding the outlier, giving place to new weights and a new pose after applying wPOSIT without the outlier. Using this as initialization, an iterative OC process is carried out. The estimated pose, initially obtained without the outlier (OD) but calculated using the corrected point in subsequent iterations (OC), is used to calculate the ideal 2D image position of the detected outlier. This is done through a pose denormalization process. Let's explain this in detail: we have previously obtained a reference template of the 2D shape in a frontal view by pose-normalizing the 2D points detected in the initial frame. This reference is supposed to be the ideal tracking and the calculated TAIs measure the 2D tracking performance with respect to that reference. Therefore, we can apply an inverse process of pose denormalization to that template and obtain the corresponding 2D projection of any of the reference points in the desired pose. This process is illustrated in Fig. 4.4.

The convergence of the iterative process is assessed by measuring the Euclidean 2D distance between the corrected outlier and its image position in the previous iteration. Once this distance is smaller than a chosen threshold, we assume that convergence has been reached and the final corrected image position for the outlier is obtained. This final image is pose-normalized with the last estimated 3D pose and the process to obtain the final TAIs is followed. Similarly to the OD method described in the previous section, if the new TAIs, calculated with the corrected outlier, exceed in average the old ones, previous to the outlier correction, the outlier is confirmed and the correction accepted. This OC loop is iteratively repeated until no more outliers are detected, checking the lowest TAI among the new ones against the threshold at each iteration. Once the outlier-corrected 2D-3D correspondences have been determined and the final TAIs have

Algorithm 4.3 wPOSIT+OD+OC

Input: $\boldsymbol{X}^p: (X^p, Y^p, Z^p)$, $\boldsymbol{x}_i^p: (x_i^p, y_i^p)$; $p = 1 \dots P$ points, $i = 1 \dots N$ frames. We denote them as $\boldsymbol{X}$ ($P \times 3$ matrix) and $\boldsymbol{x}_i$ ($P \times 2$ matrix) for simplicity.
Process initial frame (we denote it as *ref* for 'reference'):
    Apply POSIT: $(\boldsymbol{R}_{ref}, \boldsymbol{T}_{ref}) = POSIT(\boldsymbol{X}, \boldsymbol{x}_{ref})$
    Pose-normalized image points: $\tilde{\boldsymbol{x}}_{ref} = PoseNorm(\boldsymbol{R}_{ref}, \boldsymbol{T}_{ref}, \boldsymbol{x}_{ref})$
    Calculate r_{ref} and s_{ref} invariant shape metrics from $\tilde{\boldsymbol{x}}_{ref}$
for $i = 2 \rightarrow N$ **do** (process rest of frames)
    Apply POSIT: $(\boldsymbol{R}_i, \boldsymbol{T}_i) = POSIT(\boldsymbol{X}, \boldsymbol{x}_i)$
    Pose-normalized image points: $\tilde{\boldsymbol{x}}_i = PoseNorm(\boldsymbol{R}_i, \boldsymbol{T}_i, \boldsymbol{x}_i)$
    Calculate r_i and s_i invariant shape metrics from $\tilde{\boldsymbol{x}}_i$
    Obtain comparison metrics: $cr_i = f(r_{ref}, r_i)$ and $cs_i = f(s_{ref}, s_i)$
    Tolerance analysis to obtain TAI: $f_i = TolAnalysis(cr_i, cs_i)$
    while $(min(f_i) < \lambda)$ **do**
        Repeat steps 9-11 without outlier and obtain f_i^{OD}
        Calculate weights without outlier: $w_i^{OD} = WeightCalc(f_i^{OD})$
        Apply wPOSIT without outlier: $(\boldsymbol{R}_i^{OD}, \boldsymbol{T}_i^{OD}) = wPOSIT(\boldsymbol{X}^{OD}, \boldsymbol{x}_i^{OD}, w_i^{OD})$
        Initialize: $(\boldsymbol{R}_{iter}^{OC}, \boldsymbol{T}_{iter}^{OC}) = (\boldsymbol{R}_i^{OD}, \boldsymbol{T}_i^{OD})$, $d_{iter}^{OC} = \infty$
        while $(d_{iter}^{OC} < \varepsilon)$ **and** $(iter \leq M_{iter})$ **do**
            Outlier correction: $\boldsymbol{x}_{iter}^{OC} = PoseDenorm(\boldsymbol{R}_{iter}^{OC}, \boldsymbol{T}_{iter}^{OC}, \boldsymbol{x}_i^{OD}, \tilde{\boldsymbol{x}}_{ref})$
            Pose-normalized image points: $\tilde{\boldsymbol{x}}_{iter}^{OC} = PoseNorm(\boldsymbol{R}_{iter}^{OC}, \boldsymbol{T}_{iter}^{OC}, \boldsymbol{x}_{iter}^{OC})$
            Repeat steps 9-11 with $\tilde{\boldsymbol{x}}_{iter}^{OC}$ and obtain f_{iter}^{OC}
            Calculate weights for OC: $w_{iter}^{OC} = WeightCalc(f_{iter}^{OC})$
            Update pose with OC: $(\boldsymbol{R}_{iter}^{OC}, \boldsymbol{T}_{iter}^{OC}) = wPOSIT(\boldsymbol{X}, \boldsymbol{x}_{iter}^{OC}, w_{iter}^{OC})$
            Update convergence metric: $d_{iter}^{OC} = EuclDist(\boldsymbol{x}_{iter}^{OC}, \boldsymbol{x}_{iter-1}^{OC})$
        end while
        if $\left(mean(f_i^{OC}) > mean(f_i)\right)$ **then**
            Confirm and correct outlier. Update $f_i = f_i^{OC}$, $\boldsymbol{x}_i = \boldsymbol{x}_i^{OC}$
        else
            Break **while loop**
        end if
    end while
    Calculate final weights: $w_i = WeightCalc(f_i)$
    Apply wPOSIT: $(\boldsymbol{R}_i', \boldsymbol{T}_i') = wPOSIT(\boldsymbol{X}, \boldsymbol{x}_i, w_i)$
end for
Output: $(\boldsymbol{R}_i', \boldsymbol{T}_i')$; $i = 1 \dots N$ frames

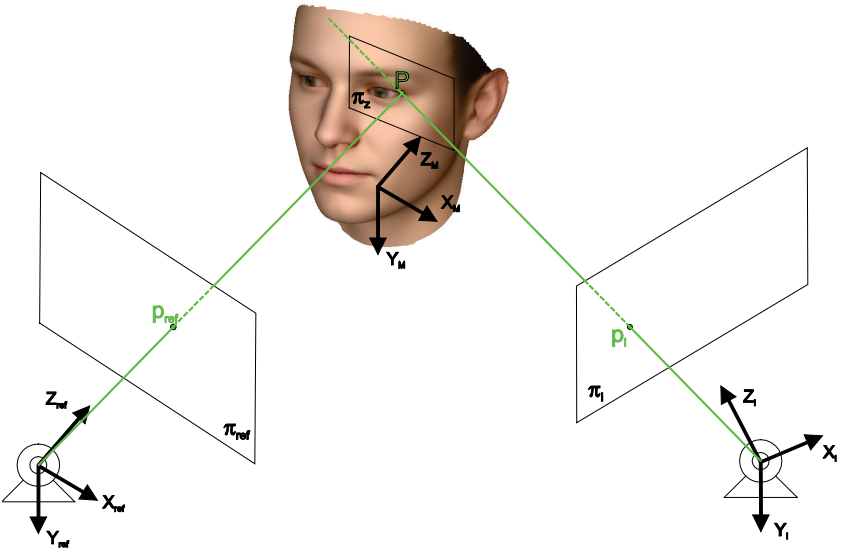

Figure 4.4: Illustration of the pose denormalization process. An image point from the reference template (p_{ref}) in the frontal-view coordinate system (X_{ref},Y_{ref},Z_{ref}) is back-projected to the model (X_M,Y_M,Z_M) and reprojected to the image plane in the current pose (π_i for (X_i,Y_i,Z_i) coordinate system).

been calculated, final weights are calculated and wPOSIT is applied in order to obtain the definitive pose for the current frame.

4.5 Application to HPE

As it has already been mentioned, all the methods presented in the previous section are devoted to enhancing the 2D tracking and 3D pose estimation of any object for which an approximate 3D model is available. However, since this thesis is framed in the specific context of HPE, this section aims to adapt the presented methods for the task and evaluate them in real conditions. Moreover, our research group works with a wider goal of developing gaze estimation methods for low resolution images, a task for which HPE is considered a necessary step, as it has already been stated in the introductory chapter of this thesis.

The process of performing HPE in a video sequence could be divided in different steps according to what has been explained in this thesis so far: 1) 2D facial feature point detection and tracking; 2) calculation of corresponding 3D points in the model; 3) 3D pose estimation using POSIT algorithm; and 4) 2D tracking and 3D pose estimation enhancement based on TAI calculation, wPOSIT application, and OD and OC processes. This section addresses the application of all these steps; first, facial feature detection and tracking is described; then, the 2D tracking and 3D pose enhancement methods described in previous sections are adapted to the specific task of HPE. The calculation of a 3D shape from 2D observations has already been addressed in detail in Chapter 3. The POSIT algorithm has already been described and characterized in Chapter 2.

4.5.1 2D Facial Feature Detection and Tracking

First of all, for HPE to be carried out in real videos, 2D facial feature point detection and tracking has to be performed. Various state-of-the-art approaches for this purpose are described and discussed in this subsection, and the approach adopted for our method is presented. Nonetheless, all the 2D tracking approaches presented here will be combined with POSIT in order to get a HPE performance that will be included in a comparison in Section 4.5.3.

4.5.1.1 IntraFace

IntraFace (IF) refers to a publicly available facial feature detection method developed by Xiong & De La Torre at the Carnegie Mellon University in 2013 [3], and recently enhanced by the same authors in [11].

Algorithm Description

IF is a supervised descent method (SDM) that detects a set of facial feature points distributed along the face based on previous training. Non-linear optimization methods are largely used in computer vision problems, typically 2nd order descent methods, due to their robustness. However, IF is based on applying a SDM for minimizing a non-linear least squares (NLS) function. A sequence of generic descent directions minimizing the function is learned in a supervised manner during training, and this sequence is then applied to detect the facial points in new input images by driving the optimization search. Xiong & De La Torre [3] showed that this method overcomes the main drawbacks of 2nd order

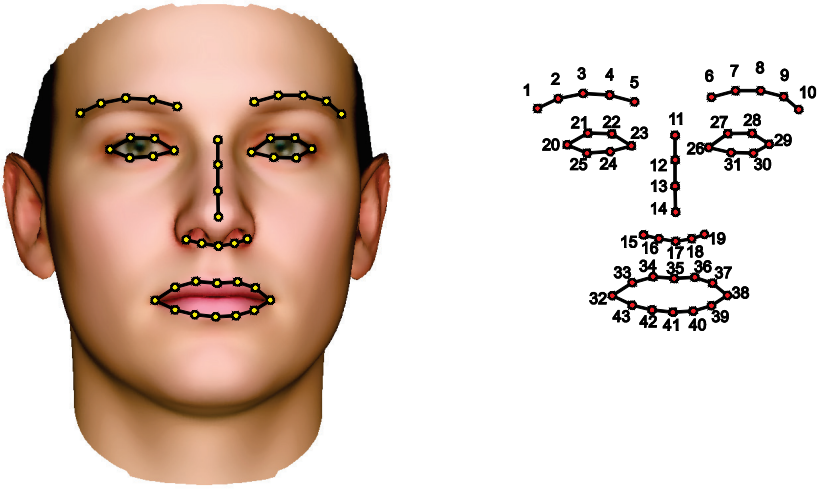

Figure 4.5: The 43-point model for the IF tracking algorithm we have implemented. The interior part of the lips is excluded from the original 49-point model in IF.

optimization schemes, i.e. the possibility of the function not being analytically differentiable, and the expensive computation of the Jacobians and Hessians.

More recently, an extension of the SDM has been proposed by the same authors [11], called Global SDM (GSDM). Depending on the objective function, standard SDM may find several local minima in a small neighborhood and therefore average conflicting gradient directions, which could result in a poor performance. GSDM extends the standard approach by dividing the search space into different regions containing similar gradient directions, which solves the previous problem and provides a more efficient minimization process.

Implementation

The IF version based on the standard SDM is publicly available for research purposes at http://www.humansensing.cs.cmu.edu/intraface/. We have downloaded the Matlab implementation of IF, which detects 49 facial feature points. This point-set includes the interior part of the lips, which we have excluded in our experiments, resulting in the 43-point model shown in Fig. 4.5. IF is initialized at each frame using the detection resulting from the previous frame, assuming a small motion between frames. In order to provide the method with a

coarse estimate of the face position in the initial frame of each video sequence, thus avoiding misdetections, the classical face detector introduced by Viola & Jones [12] has been used.

4.5.1.2 Appearance-Based Methods

"Appearance-based methods" refers to the popular ASM [4], [13] and AAM [5], [14]–[19] techniques introduced by Cootes et al. more than a decade ago that are based on statistical models of shape and appearance. They can be applied to static images in order to segment a set of key feature points defining the object of interest, in this case the face.

Algorithm Description

ASM and AAM are based on a previous learning stage using a training image set in which key segmentation features have been annotated. ASM learns the statistical behavior of the object shape and the appearance of the neighborhood of each landmark, whereas AAM learns textures in the regions between landmarks in addition to the object shape.

The training samples are images where the object has already been segmented. Segmentation consists in defining the shape of interest through the placement of several landmarks distributed along it. Each landmark must define a unique point of correspondence between all the shapes, so they must be placed carefully. Using these training samples, ASM and AAM learn the patterns of variability of the object and are able to automatically locate the landmarks. While ASM only works with shape constraints, AAM models also the texture of the object, creating a combined model from the coupled relationship between shape and texture.

Principal Component Analysis (PCA) is usually applied to the training data in order to reduce its dimensionality and obtain the modes that best describe the observed variation. Using this technique, any shape instance can be described with the following equation:

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b} \qquad (4.13)$$

where $\bar{\mathbf{x}}$ is the mean shape, $\mathbf{b} = (b_1, b_2, \dots, b_t)^T$ is a t-dimensional vector containing the set of parameters that define the statistical shape and appearance model, and $\mathbf{P} = (\mathbf{p}_1|\mathbf{p}_2|\dots|\mathbf{p}_t)$ is a matrix with the t main eigenvectors of the

covariance matrix, which are associated to the *t* highest eigenvalues. The number of modes to retain (*t*) is usually chosen as the proportion f_v (typically 95%-98%) of the total variance observed in the training we wish the model to explain. The total variance exhibited in the training is defined as the sum of all the eigenvalues λ_i. Thus *t* is calculated as the minimum integer that satisfies $\sum_{i=1}^{t} \lambda_i \geq f_v \sum_{\forall i} \lambda_i$.

In ASM, the appearance model of each landmark is usually built using fixed-size grey profiles normal to the boundary defined by the shape and centered in the landmark itself. In order to reduce the effect of global intensity changes, the normalized first order derivative of the grey level profile is used. In AAM, on the other hand, the landmarks are triangulated and the texture contained in each triangle area is modeled.

Implementation

For ASM, a Matlab implementation based on the work by Cerrolaza et al. [20] has been chosen. The algorithm works at multiple resolutions, looking for the learned shape gradually from a coarser resolution to the finest possible resolution. In the case of AAM, the compositional gradient descent method introduced by Amberg et al. [21] has been chosen in order to achieve the best fitting.

Both algorithms have been trained to perform an automatic detection of a 54-landmark model that composes the face shape for each input image. This model is defined in Fig. 4.6. The training has been carried out using the automatically labelled faces from the UPNA Head Pose Database [22] described in Chapter 2.

The initialization of the algorithm, consisting in an initial estimation of the landmarks' location, is usually critical to achieve a good fitting. In this work, both methods are initialized at each frame using the detection resulting from the previous frame, assuming a small motion between frames. To get a coarse estimate of the face position in the initial frame of each video sequence, the classical face detector introduced by Viola & Jones [12] is used.

4.5.1.3 Optical Flow

The term *optical flow* refers to the apparent motion of objects, edges and surfaces in an image due to the relative motion between the camera and the scene. One of the most widespread optical flow estimation algorithms in

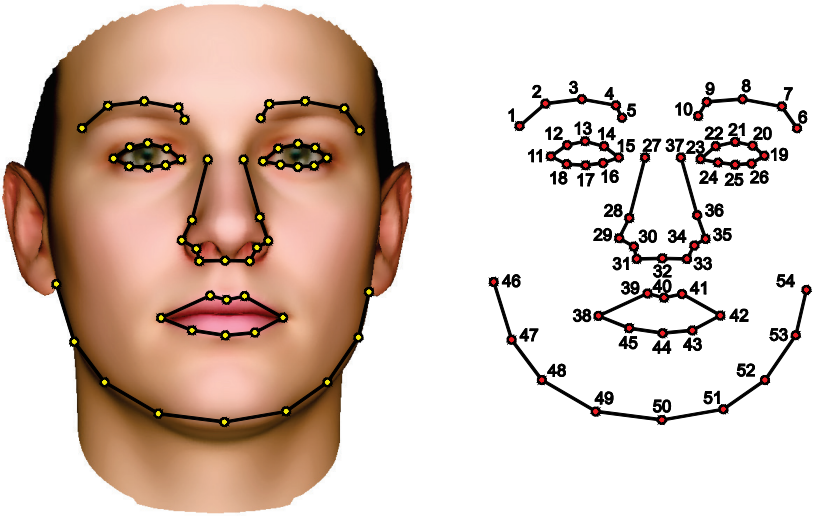

Figure 4.6: The 54-point model defined for the ASM and AAM methods that we have implemented for facial point detection and tracking.

computer vision was introduced by Lucas & Kanade in 1981 [2], and it can be applied to track a previously detected set of characteristic image points along a sequence in time.

Algorithm Description

The Lucas-Kanade algorithm is a Gauss-Newton gradient descent non-linear optimization algorithm that is based on the assumption that the displacement of the image contents between two instants close in time is small (*small motion*), and that this displacement is constant for all the pixels within a neighborhood (*spatial coherence*). This neighborhood is defined in practice as an integration window in which the match for the point from the previous image frame is looked for in the current one. Those assumptions are reasonable to make in head tracking, making the Lucas-Kanade algorithm a priori suitable for the task.

The algorithm works by searching for the point that minimizes a residual function. Let's consider two consecutive image frames, I and J, where the points

$\boldsymbol{u} = (u_x, u_y)$ and $\boldsymbol{v} = (v_x, v_y)$ are a match, such that $I(\boldsymbol{u})$ and $J(\boldsymbol{v})$ correspond to the same 3D point in the face. The points $\boldsymbol{u}$ and $\boldsymbol{v}$ are related by the 2D velocity vector $\boldsymbol{d} = (d_x, d_y)$, such that $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{d} = (u_x + d_x, u_y + d_y)$. Since the similarity between points in both images is measured as brightness constancy, this similarity has to be defined in a 2D neighborhood sense in order to overcome the *aperture problem*, which basically consists in the difficulty of determining the motion direction of a point lying on a visual edge in the image. The neighborhood is defined using an integration window of width $2w_x + 1$ and height $2w_y + 1$, and the velocity vector that determines the new position of a tracked point is defined as the one that minimizes the following residual function:

$$\varepsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \left(I(x,y) - J(x + d_x, y + d_y)\right)^2 \qquad (4.14)$$

Even though the spatial coherence assumption is helpful when tracking low-texture points or points on edges, the Lucas-Kanade method performs best when the tracked points are characteristic features, which usually means being corners. The size of the integration window is a parameter of the algorithm that presents a tradeoff between accuracy and robustness; a small window may help accurately detecting small motions, whereas a large window deals better with large displacements. One of the main advantages of the Lucas-Kanade algorithm is that it is a general-purpose method able to track any kind of object without the need of specific training.

Implementation

In order to overcome the tradeoff problem, the most typical implementation of the Lucas-Kanade method is the pyramidal scheme [23]. It consists in a coarse-to-fine registration, in which the original image is downsampled to lower resolutions building a pyramid in which each level corresponds to one image resolution. This way, the algorithm is recursively applied at each level, starting from the coarser resolution in which larger displacements can be detected using the same integration window. The outcome of each level is used as an initial guess for the next level, increasing the accuracy until the final tracking result is obtained from the original resolution image.

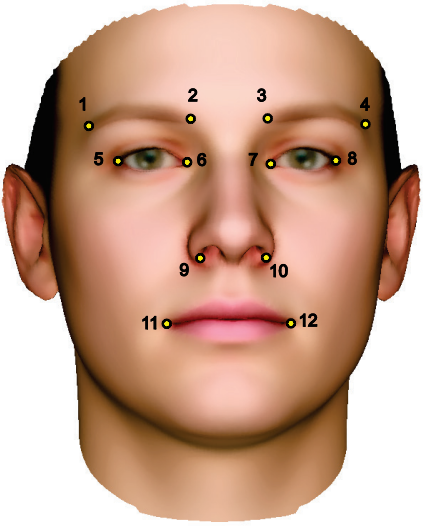

Figure 4.7: The 12-point model implemented by us in the proposed face tracking method.

Different implementations of the pyramidal Lucas-Kanade method are available, such as a C++ implementation in the OpenCV libraries or a Matlab implementation recently included in the Computer Vision System Toolbox.

4.5.1.4 Proposed Method

We have chosen to implement Lucas-Kanade as the 2D tracking method for our HPE system. The main reason for this choice is that this tracking method is universal for any kind of 3D object, whereas IF has only been trained for faces and ASM and AAM also require a specific training for each kind of object, which is a time-consuming task, often tedious because of the difficulty of finding labelled datasets. Besides, the enhancement methods based on TAI calculation presented earlier have been shown to be valid for any kind of object to track, which makes them more interesting to be combined with a universal tracking method. Moreover, the Lucas-Kanade method treats each point as an independent entity, whereas IF or ASM/AAM work with point meshes that deform jointly, keeping a high degree of coherence in interlandmark relations. Therefore, the enhancement methods based on TAI calculation seem to be better suited for Lucas-Kanade, where it is more likely to find isolated outliers.

However, as described earlier, Lucas-Kanade is a method that needs an initial set of 2D points to start tracking. We have chosen to carry out this initialization by applying ASM to the initial frame of each video sequence. As already stated in Chapters 2 and 3, we are going to work with the 12-facial-point model shown in Fig. 4.7. Note that this 12-point model is part of the 54-point model detected by our ASM implementation and shown in Fig. 4.6.

4.5.2 Tracking Accuracy Index (TAI)

This section details the adaptation of the methods based on TAI calculation presented earlier to the specific problem of HPE. The process of obtaining a 3D point set that models the head has already been described in Chapter 3, and is carried out using the algorithm of Bundle Adjustment (BA), since it is the reconstruction method that performs best according to the results presented in that chapter. The tracking of the corresponding 2D points in the image sequence is carried out using Lucas-Kanade, as stated in the previous subsection.

4.5.2.1 Invariant Shape Metrics

The two invariant shape metrics defined in equation (4.1) are calculated for all the triplets of points available from the 2D points given by Lucas-Kanade in each video frame. Since we have chosen to use the 12-point model presented earlier, we will have $\frac{12!}{3!9!} = 220$ different triplets of points. Three different r metrics and s metrics are obtained from each triplet (i.e. indexes ordered as jkl, klj or ljk), which results in 660 r and s shape metrics. This, on the other hand, results in each of the 12 points being included in or represented by 165 triplets, and thus being described by 165 r metrics and 165 s metrics.

4.5.2.2 Pose Normalization

Every frame in an image sequence is pose-normalized to a frontal position through the process described in Section 4.3.2. Using the 2D points given by Lucas-Kanade and their corresponding 3D points, POSIT is applied to obtain the current pose of the head with respect to the camera, with which the pose-normalized appearance of the tracked points is obtained. By doing this, the 2D facial appearance changes in the image due to the current 3D head pose are overcome. The two shape metrics are calculated once the face has been pose-normalized for each frame.

4.5.2.3 Subject Independency

As explained in Section 4.3.3, subject independency is achieved by using the initial frame of the sequence as template and comparing the metrics obtained in any subsequent frame with the template metrics. This way, appearance variations due to anatomical differences between subjects do not have any effect on the shape metrics.

Facial points in the initial frame are detected using ASM and the initial head pose estimated using POSIT. These points are then pose-normalized in order to build the template for comparison for the whole sequence, which thus consists of a frontal view of the facial points initially detected for the current subject. The ASM detection is assumed to be correct, and thus the template for comparison is assumed to be the pose-normalized appearance that the facial point set of any frame should have if it is being tracked accurately.

4.5.2.4 Tolerance Model

The tolerance model for HPE has been built with the aid of the UPNA Head Pose Database that contains a large set of automatically annotated faces [22]. Non-parametric tolerance intervals have been calculated using this set according to equation (4.6). Since the labelled faces represent the ideal tracking, we can obtain the training samples by following the full procedure of pose normalization and comparison–shape–metrics calculation using the automatically annotated 2D points in the database. Being the tracking ideal, the 2D error in the pose-normalized points will just be due to inaccuracies in the 3D model, which has previously been obtained through BA. Actually, this model reconstruction error leads to a HPE error, as it has been seen in Chapter 3, which in turn leads to inaccuracies in the pose normalization and in the resulting 2D observations that correspond to the frontal view of the face. Therefore, we are able to build the tolerance model in perfect tracking conditions and we can assume that any sample that lies outside these intervals corresponds to inaccurate 2D tracking.

All in all, we have made use of a large dataset (36.000 frames) of labelled training faces to calculate non-parametric tolerance intervals for the two comparison shape metrics for every combination of triplets of points with the 12-point model, i.e. 660 intervals T_{cr}^{jkl} and T_{cs}^{jkl}, as defined in Section 4.3.4. Actually, the tolerance intervals have been calculated separately for each of the ten subjects in the database, and then averaged to obtain the final tolerance model.

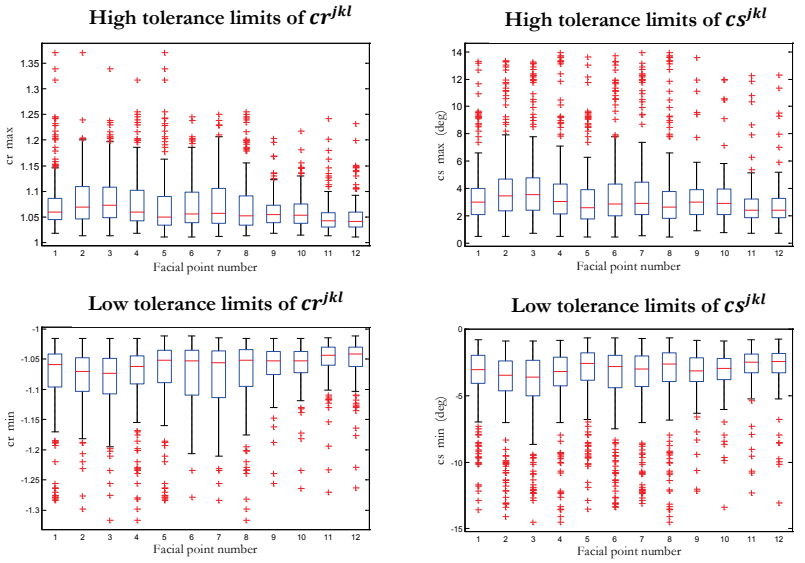

Figure 4.8: Boxplot showing the obtained tolerance model for cr and cs comparison shape metrics, calculated from the 2D ground truth of the UPNA Head Pose Database.

Fig. 4.8 shows the resultant tolerance model, represented as boxplots showing the high and low limits of the tolerance intervals of the comparison metrics (cr^{jkl} and cs^{jkl}) for each of the twelve facial points. That is to say that, for each facial point, the 165 tolerance intervals in which that point is involved, defined as $T_c^{jkl} = [min(c^{jkl}),\ max(c^{jkl})]$, and corresponding to each comparison metric, are represented. Note that each interval is represented three times in the boxplot, one for each point of the triplet to which that tolerance interval corresponds. This can be noticed looking at the largest tolerance intervals, represented by the most extreme high and low limits in the boxplot.

4.5.2.5 Weight Calculation

Using this tolerance model, the TAI f_i^p of each point p in each frame i can be obtained for the 12-point model following equations (4.8), (4.9) and (4.10). As described in Section 4.3.5, the TAI consists of two contributions (i.e. fr and fs), one from each type of shape metric, and each of them consists in turn of the

contributions of 165 metrics (i.e. cr^{jkl} and cs^{jkl}) from all the triplets in which the point is involved.

We have developed a method to rank the metrics associated to each point according to their sensitivity to the tracking and HPE accuracy. This ranking allows us to select only a certain number of metrics for the calculation of the TAI of each point by choosing the most sensitive ones to tracking and HPE inaccuracies. This is especially useful when very large numbers of metrics are available, since the calculation may be time consuming and over-fitting problems may also arise. The 12-point model chosen in this work gives rise to 660 total metrics (165 per point), but note that, for instance, a 20-point model would result in 3.420 total metrics (513 per point), and a 50-point model would generate 58.800 total metrics (3.528 per point). Hence, the method that is going to be presented may be especially necessary when working with this kind of face characterization models.

The ranking method can be divided in two parts, resulting in two independent metric orders that are eventually combined in order to obtain the final ranking. The first part tries to account for the 2D tracking inaccuracy effect. For this purpose, the pose-normalized faces used for building the tolerance model as described in the previous subsection are taken to build the group of inliers, i.e. the group of metrics corresponding to accurately tracked points. Having those pose-normalized faces as reference, the group of outliers is built by taking a window around each of the points, which represents inaccurately-tracked points. It is also possible to be less restrictive by choosing a small inner window around each point to build the group of inliers and a bigger outer window to build the group of outliers. We can then calculate the intersection between inlier and outlier metrics, and rank the metrics from the one with the smallest average intersection (i.e. more sensitivity to tracking inaccuracies) to the one with the highest intersection (i.e. less sensitivity to tracking inaccuracies). The second part, on the other hand, tries to account for the effect of HPE inaccuracies on the pose-normalized faces. As it has been described earlier, errors in the 3D model reconstruction result in HPE inaccuracies, which in turn lead to errors in the pose-normalized 2D facial points. By looking at the variance of the metric values calculated from these pose-normalized faces, we can rank the metrics from less sensitive to HPE errors (smaller variance) to more sensitive (bigger variance). We can thus build a final ranking of metrics that accounts for their sensitivity to both 2D tracking and 3D HPE inaccuracies by combining the two rankings and obtaining the average position for each metric. We have chosen in this thesis to

work with the first 100 metrics in the ranking for each point, based on preliminary experimental results.

Once the TAI for each point in each video frame has been calculated following the procedure described, weights for wPOSIT are obtained using equation (4.11).

4.5.2.6 Outlier Detection (OD) and Correction (OC)

OD and OC can be applied to head tracking and pose estimation as described in Algorithms 4.2 and 4.3 without any particular modification. When applying OD, it must be taken into account that POSIT needs at least four point correspondences in order to estimate a 3D pose, so there is a limit in the number of outliers that can be detected in one particular frame. Regarding OC, the pose denormalization process is applied as described in Section 4.4.2 with the aid of the 3D face model reconstructed using BA and the 2D points detected by ASM for the initial frame of the video sequence.

4.5.3 Results and Discussion

Final results for the HPE methods introduced in this chapter are presented in this section. For the evaluation of the methods, two head pose databases of videos have been employed: the BU Headtracking Database [24] and the UPNA Head Pose Database [22]. The former contains 45 videos of 200 frames of 5 users performing free head movements, whereas the latter contains 120 videos of 300 frames of 10 users performing both guided and free movement sequences. HPE results have been evaluated in both databases. In addition to that, the UPNA database also contains a 2D ground truth (see Chapter 2), which allows us to evaluate 2D tracking performance. On the other hand, the BU database has been widely used for evaluation of HPE methods in the last decade, which gives us the opportunity to compare our methods with the state of the art.

The methods proposed in this thesis give room to four different HPE approaches that are going to be referred to as: 1) the basic approach; 2) wPOSIT; 3) wPOSIT+OD; 4) wPOSIT+OD+OC. The basic approach applies POSIT to the tracked points, as described in Section 4.1. wPOSIT is described in Algorithm 4.1 and consists in calculating a TAI in order to assign weights to POSIT. wPOSIT+OD integrates an iterative outlier detection module together with the wPOSIT method, as described in Algorithm 4.2. Finally, wPOSIT+OD+OC is described in Algorithm 4.3, and it consists in adding an outlier correction module

to the previous method. These four approaches are evaluated in this section, both for 2D tracking accuracy and 3D HPE accuracy. Besides those methods, the ASM, AAM and IF 2D tracking methods presented in Section 4.5.1 are included as reference in a performance comparison, combined with POSIT in order to obtain additional HPE results. In addition to that, various state-of-the-art HPE algorithms that have reported results for the BU database are also included in the comparison.

4.5.3.1 Tracking Results

2D facial point tracking results for the UPNA Head Pose Database are shown in Tables 4.1 and 4.2. Table 4.1 presents an exhaustive comparison between the four methods introduced in this chapter, whereas Table 4.2 presents an accuracy comparison between these four approaches and two state-of-the-art methods used as reference. The 2D facial point tracking accuracy is measured as the average Euclidean distance between the tracked points and the 2D ground truth provided with the database, which was obtained through an automatic facial annotation method described in Chapter 2. No tracking results are shown for the BU Headtracking Database because no such 2D ground truth is available for those image sequences.

Table 4.1 shows a detailed comparison between the approaches proposed in this thesis, in which the errors have been measured for all the frames (AF) in the database, on the one hand, and for frames in which at least one outlier has been detected (OF), on the other hand. The proportion of frames with at least one outlier is also shown (OF/AF). This proportion depends on the parameters of the algorithm, especially the threshold for outlier detection. These parameters have been optimally adjusted in preliminary experimentation, and it results in 23,52% of the frames in the database containing at least one outlier according to our OD module. The basic approach is the reference for comparison because it is the tracking given by Lucas-Kanade for the 12 facial points. The wPOSIT approach gives the same 2D tracking results as the basic approach, since it affects the 3D HPE by giving weights to the tracked points based on the TAI, but does not alter the tracked 2D points themselves. The OD approach eliminates outliers in each frame independently, so the tracking error is obtained by averaging the deviation of the remaining inliers with respect to the 2D ground truth. The OD+OC approach corrects those outliers, which means that the tracking error is obtained with the 12 points in every frame once the outliers have been corrected.

Table 4.1: 2D head tracking accuracy results for the UPNA database (1280×720 resolution). An exhaustive comparison between the four approaches presented in this thesis is shown, including errors for all frames (AF) and only for frames in which at least one outlier is detected (OF). The 2D error is measured as the average Euclidean distance between the tracked points and the 2D ground truth. The accuracy gain given by each method with respect to the basic approach is also shown, as well as the proportion of frames with at least one outlier.

Method	Tracking Error (px)		Gain Obtained (%)		OF/AF (%)
	AF	OF	AF	OF	
Basic Approach	4,03	6,83	-	-	-
wPOSIT	4,03	6,83	-	-	-
wPOSIT+OD	3,56	4,83	11,69	29,28	23,52
wPOSIT+OD+OC	3,70	5,44	8,14	20,38	23,52

Table 4.2: 2D head tracking accuracy results for the UPNA database. Comparison between the proposed approaches and state-of-the-art tracking methods for all frames in the database.

Method	Tracking Error (px)
Basic Approach	4,03
wPOSIT	4,03
wPOSIT+OD	3,56
wPOSIT+OD+OC	3,70
ASM	5,89
AAM	6,70

It is observed that the basic approach obtains an error of around 4 pixels for all frames, and 6.8 pixels for only those frames with at least one outlier, for a 1280×720 video resolution. This error is reduced to 3.70 and 5.44 pixels with OD+OC, and even further to 3.56 and 4.83 pixels with just OD. This shows that the outliers are indeed correctly detected, because removing them or correcting them actually reduces the 2D tracking error. The fact that the OD approach achieves better results than the more promising OD+OC approach implies that, even if the OC is working (it improves accuracy with respect to the basic approach), the corrected outliers are still slightly less accurate than the rest of the points (the inliers), and therefore the average error increases slightly with respect to the approach where the outliers are simply removed. However, it is important to note that this does not necessarily mean that the same behavior is to be expected in HPE, where the number of points available for the pose estimation

has certain significance, whereas it does not have any effect on the average 2D tracking error calculation. This point will be further developed in the next section.

In terms of the accuracy gain given by the enhanced approaches, it is observed that the OD and OD+OC methods achieve a 11.69% and 8.14% tracking accuracy improvement respectively if the 36.000 frames of the database are taken into account, and a 29.28% and 20.38% improvement if we compare the tracking only for frames where outliers have been detected. All in all, we can expect a tracking accuracy improvement of more than 10% using the OD module in normal 2D tracking conditions, and an accuracy improvement of up to 30% in noisy tracking conditions (all frames containing at least one outlier).

Table 4.2 presents a comparison of our methods' performance against two other state-of-the-art 2D tracking methods, i.e. ASM and AAM. IF has not been included in this comparison, since the facial points detected by the algorithm do not exactly correspond anatomically with the annotated 2D ground truth for the UPNA Head Pose Database. ASM and AAM have been trained and evaluated with images from our database on a *leave one out* basis. The comparison shows that our tracking approaches clearly outperform both ASM and AAM, which are more than 2 pixels less accurate than our best option, i.e. wPOSIT+OD. ASM and AAM are widely accepted methods for tracking purposes in computer vision applications, and show good performance if they are correctly trained and initialized, which we have achieved through the leave one out scheme and the Viola-Jones face detector respectively. Therefore, Table 4.2 shows the validity of the approaches presented in this thesis for 2D face tracking and the results seem encouraging for the goal of performing HPE, which will be studied in detail in the next section.

4.5.3.2 HPE Results

HPE results for both the UPNA Head Pose Database and the BU Headtracking Database are presented in this section. We will mainly focus on rotation errors, since that is what most methods in the literature report. However, translation errors will also be briefly introduced at the end of this section. Regarding rotation errors, Table 4.3 and Table 4.4 present an exhaustive comparison between the four HPE methods proposed in this chapter for the UPNA and BU database respectively, whereas Table 4.6 and Table 4.7 present a HPE accuracy comparison between these four approaches and other methods of reference in the literature for the UPNA and BU database respectively. A comparison regarding the computational cost of the proposed approaches is

shown in Table 4.5, and translation errors for the UPNA database are presented in Table 4.8.

Three different HPE rotation errors have been obtained, namely *est-gt*, *est0-gt* and *est0-gt0*. The first error, *est-gt*, is calculated as the average of the absolute difference between the estimation and the 3D ground truth across all the video frames in the database. The second error, *est0-gt*, is calculated as the mean absolute difference between the 3D ground truth and a *zeroed* estimation. This estimation consists in calculating the transformation that needs to be applied to the pose in the initial frame of each video in order to get a zero rotation, which is given by the inverse of the rotation matrix estimated in the initial frame. This transformation is then applied to the rest of the frames of the video. The third error, *est0-gt0*, is given by the mean absolute difference between the zeroed estimation and the zeroed 3D ground truth, obtained by applying the same transformation process to the original 3D ground truth.

The original *est-gt* compares the raw estimation given by our method with the original ground truth. The zeroed estimation (*est0*) is as valid as the non-zeroed one, since no ground truth data is used for the transformation, meaning that the transformation can be performed in any real situation where no ground truth is available. Of course, it only makes sense when an initial frontal face can be assumed, which is often the case (as in the UPNA and BU databases). Although the initial rotation given by the 3D ground truth will not be exactly zero, it has been observed that the inaccuracy of the reconstructed 3D head model introduces a systematic pose estimation error bigger than the deviation from zero of the initial 3D ground truth. We know this HPE error in the initial frame to be mostly due to the 3D head model inaccuracy because the 2D landmark detection in the initial frontal face has been observed to be highly accurate, giving a 2D average error for the UPNA database of 2.32 pixels. Learning the transformation that zeroes the estimation in the first frame allows applying the correction in the rest of the video. HPE results in Tables 4.3 and 4.4 show that this transformation indeed helps correcting partly the systematic error introduced by the inaccuracy of the reconstructed 3D head model, since the HPE error is considerably reduced in all cases. The third error aligns the estimation and the 3D ground truth at the beginning of each video by applying a transformation that zeroes both. This can be defined as a differential HPE error, which is the one that best describes the performance of a head pose estimator in a real application, where we have an estimation method and no ground truth. In such a situation, the frontal pose in which the estimator will give a zero-rotation must be defined

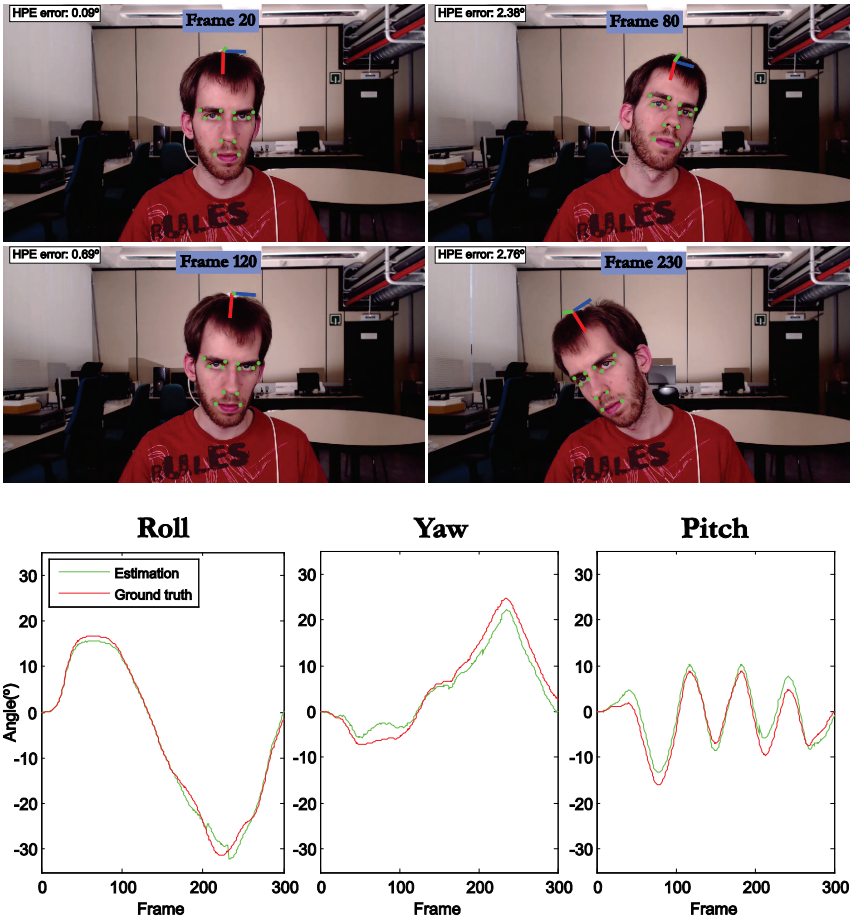

Figure 4.9: Visual HPE results on one of the videos from the UPNA database. Four different frames are shown on top, with the tracked points superimposed to the image and the three spatial axes representing the estimated pose on top of the head. Instantaneous average HPE errors are also shown. Below, estimation and ground truth curves for roll, yaw and pitch are represented for the same video sequence.

by the user. This should be done in a previous calibration step, where the user would be asked to face the camera in order to set the frontal reference position, acquiring the head pose given by the estimator at that moment. This result would then be used to calculate the transformation that will be applied to every frame after, therefore setting the zero and what we consider the exact frontal pose. From that moment on, the accuracy of the head pose estimator will be defined

just by the differential pose estimation, since the absolute pose has been calibrated by the user.

Fig. 4.9 and Fig. 4.10 show visual results on the UPNA and BU database respectively. Four frames corresponding to four different instants in one of the videos of each database can be observed in each of the figures, with the tracked points superimposed to the face and the three spatial axes representing the estimated head pose on top of the head. The three rotation curves are shown below in both figures, where the estimated angle and the ground truth have been plotted for comparison. Both example videos contain large head rotations along the three axes and show in a visual manner the accuracy of the presented method in 2D tracking and 3D HPE.

Table 4.3 and Table 4.4 show a detailed comparison between the approaches proposed in this thesis for the UPNA and BU database respectively, in which HPE errors have been measured for all the frames (AF) in the database, on the one hand, and for frames in which at least one outlier has been detected (OF), on the other hand. The proportion of frames with at least one outlier is also shown (OF/AF). The basic approach is the reference for comparison, and the gain given by each of the other approaches for AF and OF is presented in order to assess the improvement achieved through the enhancements proposed. This gain has been calculated for the differential HPE error in all cases, since it is the error that represents best the real performance of the algorithm.

In the case of the UPNA database, we observe a clear improvement as we go from one method to the next, and the behavior is similar for the three HPE errors. Looking at the differential error, we see that we start from average errors of 1.30º and 2.67º for AF and OF respectively with the basic approach, and improve to 1.21º and 2.30º when we apply weights with wPOSIT, 1.18º and 2.18º when we use the OD module and, finally, 1.13º and 1.97º when we incorporate the OC module and use the complete method. The proportion of frames in which at least one outlier has been detected is the same as for the tracking results presented in the previous section, i.e. 23.52%. Note that, regarding the 2D tracking results, we have previously observed that the wPOSIT+OD method obtains a higher accuracy than the wPOSIT+OD+OC method. However, as we have stated before, the number of points available for the pose calculation is important, and we can observe that correcting the outliers indeed leads to a more accurate 3D pose estimation.

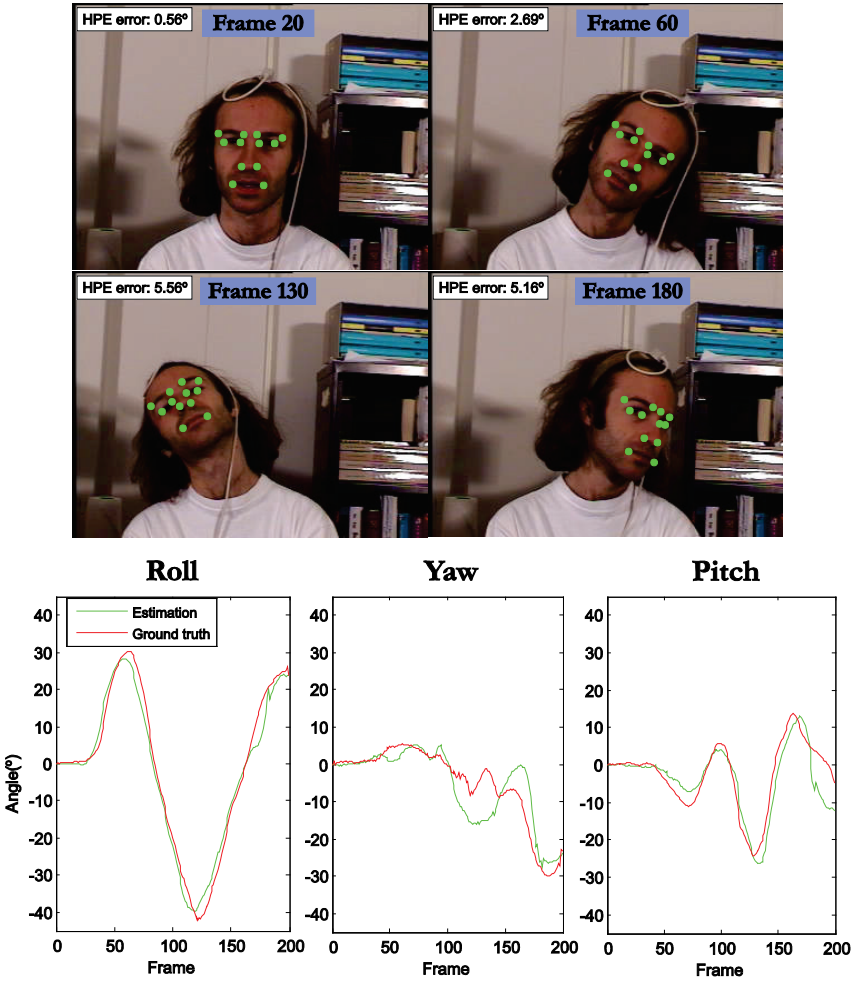

Figure 4.10: Visual HPE results on one of the videos from the BU database. Four different frames are shown on top, with the tracked points superimposed to the image and the three spatial axes representing the estimated pose on top of the head. Instantaneous average HPE errors are also shown. Below, estimation and ground truth curves for roll, yaw and pitch are represented for the same video sequence.

In terms of gain with respect to the basic approach, implementing the TAI estimation and weight application to POSIT gives an improvement of 7.42% and 13.73% for AF and OF respectively, incorporating an OD method results in an improvement of 9.69% and 18.44%, and using the full wPOSIT+OD+OC method we achieve the highest accuracy gain with 13.38% and 26.09%. These

Figure 4.11: Visual example of the effect of the OC module on a frame from the UPNA database. Inliers are represented in green, the outlier detected by the OD module in red, and the corrected point estimated by the OC module in blue. Instantaneous HPE errors of the four approaches are shown on the top left side.

numbers demonstrate the validity of the approaches proposed, leading to an improvement of almost 15% in normal tracking conditions and over 25% in noisy tracking conditions (all the frames contain outliers) with respect to the basic approach presented. In absolute terms, a differential HPE error slightly over 1° is obtained with the complete approach, which we consider a significantly high accuracy that shows the success of the proposed methods.

A visual example of the effect of the OC module is shown in Fig. 4.11, in which an example frame from one of the videos in the UPNA database is shown. The tracked inlier points are represented in green, the detected outliers are shown in red, and their corresponding positions after OC are shown in blue. The HPE error for that specific frame given by each of the approaches using their corresponding point configuration is also shown on the top-left of the image (i.e. the basic approach and the wPOSIT approach make use of both green and red points, the wPOSIT+OD approach makes use of just green points, and the wPOSIT+OD+OC approach makes use of green and blue points).

Table 4.3: HPE results for the UPNA database. An exhaustive comparison between the four approaches presented in this thesis is shown, including errors for all frames (AF) and only for frames in which at least one outlier is detected (OF). The three different HPE errors introduced in the thesis are included in the comparison. The accuracy gain given by each method with respect to the basic approach is also shown for the *est0-gt0* differential error, as well as the proportion of frames in which at least one outlier has been detected.

HPE Error – UPNA Head Pose Database

METHOD	Error	All frames (AF)				Outlier frames (OF)				Gain Obtained (%)		OF/AF (%)
		Roll (°)	Yaw (°)	Pitch (°)	Avg (°)	Roll (°)	Yaw (°)	Pitch (°)	Avg (°)	AF	OF	
Basic Approach	*est-gt*	1,35	2,42	5,83	**3,20**	2,01	4,21	6,12	**4,11**	-	-	-
	est0-gt	1,19	1,99	2,02	**1,73**	1,84	3,84	2,96	**2,88**			
	est0-gt0	0,70	1,57	1,64	**1,30**	1,53	3,69	2,78	**2,67**			
wPOSIT	*est-gt*	1,32	2,32	5,74	**3,13**	1,87	3,71	5,86	**3,81**	7,42	13,73	-
	est0-gt	1,16	1,87	1,92	**1,65**	1,69	3,29	2,64	**2,54**			
	est0-gt0	0,67	1,42	1,53	**1,21**	1,37	3,08	2,45	**2,30**			
wPOSIT+OD	*est-gt*	1,28	2,18	5,71	**3,06**	1,71	3,10	5,71	**3,51**	9,69	18,44	23,52
	est0-gt	1,11	1,77	1,96	**1,61**	1,49	2,86	2,80	**2,38**			
	est0-gt0	0,63	1,33	1,58	**1,18**	1,20	2,71	2,62	**2,18**			
wPOSIT+OD+OC	*est-gt*	1,26	2,13	5,69	**3,03**	1,62	2,91	5,63	**3,39**	13,38	26,09	23,52
	est0-gt	1,09	1,71	1,90	**1,57**	1,41	2,60	2,56	**2,19**			
	est0-gt0	0,61	1,27	1,52	**1,13**	1,10	2,43	2,38	**1,97**			

Table 4.4: HPE results for the BU database. An exhaustive comparison between the four approaches presented in this thesis is shown, including errors for all frames (AF) and only for frames in which at least one outlier is detected (OF). The three different HPE errors introduced in the thesis are included in the comparison. The accuracy gain given by each method with respect to the basic approach is also shown for the *est0-gt0* differential error, as well as the proportion of frames in which at least one outlier has been detected.

HPE Error – BU Headtracking Database

METHOD	Error	All frames (AF)				Outlier frames (OF)				Gain Obtained (%)		OF/AF (%)
		Roll (°)	Yaw (°)	Pitch (°)	Avg (°)	Roll (°)	Yaw (°)	Pitch (°)	Avg (°)	AF	OF	
Basic Approach	*est-gt*	3,38	3,97	7,81	**5,05**	4,14	10,12	8,14	**7,47**	-	-	-
	est0-gt	1,94	3,58	3,05	**2,86**	3,09	10,02	5,24	**6,11**			
	est0-gt0	1,96	3,63	2,97	**2,85**	3,10	10,40	5,24	**6,25**			
wPOSIT	*est-gt*	3,31	3,83	7,70	**4,95**	3,89	9,43	7,60	**6,97**	4,91	9,76	-
	est0-gt	1,84	3,37	2,94	**2,72**	2,78	9,11	4,64	**5,51**			
	est0-gt0	1,87	3,41	2,85	**2,71**	2,82	9,44	4,64	**5,64**			
wPOSIT+OD	*est-gt*	3,27	3,66	7,68	**4,87**	3,45	7,46	7,30	**6,07**	8,07	25,60	8,70
	est0-gt	1,82	3,20	2,89	**2,64**	2,53	7,10	4,08	**4,57**			
	est0-gt0	1,85	3,23	2,80	**2,62**	2,56	7,35	4,05	**4,65**			
wPOSIT+OD+OC	*est-gt*	3,30	3,68	7,64	**4,87**	3,76	7,71	6,90	**6,12**	7,37	22,88	8,70
	est0-gt	1,84	3,21	2,90	**2,65**	2,79	7,31	4,18	**4,76**			
	est0-gt0	1,87	3,24	2,80	**2,64**	2,80	7,60	4,06	**4,82**			

Quite a similar behavior to the one described for the UPNA database can be observed in the results for the BU database shown in Table 4.4, with slight differences. The most noteworthy one is that using the OC module does not improve further the performance of the wPOSIT+OD method, but slightly worsens it (i.e. 2.62º vs 2.64º in average for the differential error). This a priori unexpected behavior can be explained by the low quality of the videos and the inherent difficulty of tracking. A good outlier correction depends on a good 2D estimation of the rest of the points so that the 3D pose can be reasonably well estimated despite the wrongly tracked point and the corrected location of the outlier can be calculated with accuracy, thus enhancing the HPE. This is probably the scenario of many of the frames that contain outliers in the UPNA database, where the video resolution is higher (1280×720 vs 320×240) and the image acquisition conditions are better. However, the lower quality of videos in the BU database may possibly cause a worse global tracking, with the whole 2D point set less accurately estimated. This idea is enforced by the fact that we have a HPE average error more than twice as high as for the UPNA database (2.62º vs 1.13º). In those conditions, it may often be a better option to remove a detected outlier instead of trying to correct it, since inaccuracies in the tracking of the rest of the points and the subsequent pose estimation may lead to a poor outlier correction. Nevertheless, the complete HPE method (including OC) still performs far better than just the wPOSIT algorithm, which suggests that OC is working reasonably well even if it does not improve the wPOSIT+OD option.

Looking again at the differential error, the basic approach presents average errors of 2.85º and 6.25º for AF and OF respectively, wPOSIT reduces the errors to 2.71º and 5.64º, wPOSIT+OD achieves the best performance with 2.62º and 4.65º, and wPOSIT+OD+OC achieves slightly worse results with 2.64º and 4.82º. In terms of gain with respect to the basic approach, implementing the TAI estimation and wPOSIT gives an improvement of 4.91% and 9.76% for AF and OF respectively, incorporating an OD method results in an improvement of 8.07% and 25.60%, and incorporating the final OC module we achieve an improvement of 7.37% and 22.88%. Although the absolute errors are more than twice as high as those obtained for the UPNA database, the results in terms of gain given by each approach are comparable. The proportion of frames in which at least one outlier has been detected is of 8.70%, lower than for the UPNA database. This is due to the lower quality of the videos and the consequent poorer tracking; a less accurate global tracking makes it more difficult for the OD method to isolate specific outliers, since many frames will present a globally noisy tracked point set in which the best option may be to just apply weights to POSIT.

Table 4.5: HPE computation time results of the four approaches over the UPNA database, given as average values per frame.

HPE Computation Time Comparison

METHOD	Processing time (ms/frame)	Frame rate (FPS)
Basic Approach	41,85	23,89
wPOSIT	44,45	22,50
wPOSIT+OD	47,12	21,22
wPOSIT+OD+OC	48,86	20,47

In these conditions, the parameters of the algorithm are adjusted so that the OD module is less sensitive, resulting in only 8.70% of the frames containing outliers.

We can also observe in both databases that, regarding the three rotation angles, roll is clearly the most accurately estimated one, followed by yaw and pitch in the UPNA database, and pitch and yaw in the BU database. Roll estimation typically gives better results, since it consists in an in-plane rotation (i.e. rotation along a plane parallel to the image plane), whereas yaw and pitch are out-of-plane rotations which, on the one hand, may easier cause partial occlusions that affect the tracking and, on the other hand, make the estimation more dependent on the reliability of the reconstructed 3D model.

Processing times for the four HPE methods are presented in Table 4.5 in order to show the computational cost of each approach. The times have been measured as the average time employed to process each frame of the UPNA database. The processing is performed using Matlab on an IntelCore i5 PC with 6GB of RAM. The basic approach works at almost 24FPS, and the frame rate decreases progressively, down to 20-21FPS for the complete wPOSIT+OD+OC approach. The computational cost of the enhancements proposed is thus small, and we have observed that the 2D feature tracking Lucas-Kanade algorithm takes most of the time of processing (41,60ms per frame in average). The complete HPE system works almost real-time in Matlab, and thus it should be straightforward to obtain a C++ version of the algorithm for real-time video processing. Therefore, these results show the feasibility of the methods proposed in terms of computational cost.

Another issue worth mentioning is that, in order to estimate the head pose for the BU database, we need to determine the intrinsic camera parameters in advance. POSIT requires the focal length and principal point of the camera in order to produce a head pose estimate, and these parameters are not given with the BU database, which was one of the reasons for the creation of the new UPNA database. There are multiple possibilities for the estimation of these parameters. We have chosen to use a simple method for the focal length estimation, and to assume the ideal case in which the image center is the principal point. The method for the estimation of the focal length consists in a simple iterative minimization process: we project the 12 facial points of the generic BFM (see Chapter 3) on the first frame of each video, using the head pose ground truth and a certain focal length in the projection function, and we measure the distance from these projections to manual annotations of the same facial points in the image. The final error is calculated as an average of these distances. The focal length is thus adjusted in each iteration until this error is minimized. The size of the projected 3D head has a determinant effect on the focal length estimation and, since we have no way of determining the real dimensions of the heads in the BU database, it seems fair to use an average 3D head model for this purpose (i.e. the generic BFM). The resultant focal length for the BU database, obtained through the process described, is 485 pixels.

Nevertheless, it must be pointed out that moderate variations in the focal length have little effect on the estimation of the head rotation, and therefore the accuracy in the focal length estimation is not critical (it is of course for head translation estimation). We have tested this on both the UPNA and BU databases by varying the focal length around the optimal value and measuring its effect on the estimated head rotation. In both cases, variations of ±20% in the focal length produce an increase of 0.05° in the average head rotation error for the whole database. This is an important result to take into account when comparing HPE methods that require calibrated images with others that do not, since it means our method worsens its performance very little when camera calibration parameters are not available and must be calculated from the images, which involves a certain estimation error.

Table 4.6 shows a comparison of our four methods with other algorithms of reference that we have implemented and tested in the UPNA database. The comparison is carried out for the *est0-gt0* differential HPE error in order to show the performance that would be expected from each method in a real application. Since the database is very recent, there are no HPE methods in

Table 4.6: HPE results for the UPNA database. Comparison between the four approaches proposed in this thesis and other reference methods. The differential HPE error is shown in all cases.

HPE Comparison – UPNA Head Pose Database

METHOD	Error (°)			
	Roll	Yaw	Pitch	Avg
Basic Approach	0,70	1,57	1,64	**1,30**
wPOSIT	0,67	1,42	1,53	**1,21**
wPOSIT+OD	0,63	1,33	1,58	**1,18**
wPOSIT+OD+OC	0,61	1,27	1,52	**1,13**
ASM + POSIT	0,88	2,50	2,77	**2,05**
AAM + POSIT	1,04	1,63	2,19	**1,62**
IF + POSIT	0,45	0,94	1,39	**0,93**
IF pose	0,92	3,12	2,98	**2,34**

the literature that report results for it yet. Therefore, the reference methods included in the comparison result from combining the different 2D-tracking state-of-the-art algorithms described in Section 4.5.1 with POSIT, so that a 3D head pose is obtained in the output. These tracking methods are IF, ASM and AAM. Besides, the IF implementation described in Section 4.5.1.1 provides its own pose estimation through a process of fitting a 3D model to the 2D observations, and this result has also been included in the comparison. IF is an uncalibrated HPE method, meaning it does not require the camera parameters in order to produce a head pose estimate, but estimates them from the images. However, as we have stated above, moderate variations in the focal length (±20%) cause a very small increase in the HPE error (+0.05°), reinforcing the idea of a direct comparison of IF's own pose estimation with the rest of the methods in Table 4.6.

The table shows that our four proposals beat any of the appearance-based methods (ASM or AAM) combined with POSIT, as well as IF's own estimation, by a difference of between 0.5° and 1° in the case of our full method (wPOSIT+OD+OC). The HPE comparison results for the ASM and AAM are in accordance with the 2D tracking results presented in the previous section, showing that a less accurate tracking leads to a poorer HPE. The best HPE is

achieved by the combination of IF's 2D tracking with POSIT, showing a high accuracy with an average error of 0.93°, 0.2° better than the proposed method. However, our method presents some advantages over IF; while the latter has been specifically trained for face tracking, our method can be applied to any kind of 3D object without the need of training, which provides an enormous flexibility and ease of use. Besides, training processes are very time consuming and require labelled samples, which is usually a tedious task and may condition the performance of the algorithm if the labels do not accurately correspond with each other.

Table 4.7 compares our methods with the state of the art on the BU database, for which many articles in the literature have reported HPE results and thus allow a consistent comparison. Some of these methods make use of camera calibration parameters for HPE and some do not and, as it has already been mentioned, these parameters are not given for the BU database and must be estimated from images. However, the little effect that focal length variations have over HPE errors makes this comparison fair. We have also included results obtained by combining IF with POSIT, as well as IF's own pose estimation. ASM and AAM have been excluded from the comparison because they show a very poor performance on the BU database. The reason is that we have trained them using the automatic annotations from the UPNA database, and these training images show very different characteristics if compared with the BU database videos, which causes the ASM and AAM to provide inaccurate detections in many frames. Most important state-of-the-art HPE methods, developed mainly in the last decade, have been included in Table 4.7, showing that our method is the most accurate one among all. The wPOSIT+OD implementation provides the best results (2.62° average error), followed by the wPOSIT+OD+OC implementation (2.64°). Combining IF with POSIT gives the third best result (2.70°), almost identical to our wPOSIT implementation without outlier handling (2.71°). This confirms the good tracking given by IF, as shown earlier for the UPNA database, although it does not get better results than our method for the noisier BU database. The fifth best result is obtained by Wang et al. [25] (2.77°), with a method that learns keypoints that are invariant to head pose, facial expression and lighting changes, performs keypoint matching and finally solves a set of 3D-2D correspondences for the best 3D pose. This result is followed closely by the method of Xiao et al. [26] (2.80°), in which templates of the head image and the corresponding 3D pose were dynamically updated along the tracking, re-registering a new image if the pose was found to be close to one of

Table 4.7: HPE results for the BU database. Comparison between the four approaches proposed in this thesis and other reference methods. The differential HPE error is shown in all cases.

HPE Comparison – BU Headtracking Database

METHOD	Error (°)			
	Roll	Yaw	Pitch	Avg
wPOSIT+OD	1,85	3,23	2,80	**2,62**
wPOSIT+OD+OC	1,87	3,24	2,80	**2,64**
IF + POSIT	1,76	3,25	3,10	**2,70**
wPOSIT	1,87	3,41	2,85	**2,71**
Wang et al. 2012 [25]	1,86	3,75	2,69	**2,77**
Xiao et al. 2002 [26]	1,40	3,80	3,20	**2,80**
Basic Approach	1,96	3,63	2,97	**2,85**
Lefèvre & Odobez 2009 [27]	2,00	4,40	3,30	**3,23**
Prasad & Aravind 2010 [28]	3,60	3,80	2,50	**3,30**
IF pose	2,02	3,85	4,06	**3,31**
Jang & Kanade 2008 [29]	2,10	4,60	3,70	**3,47**
Asteriadis et al. 2014 [30]	2,61	4,29	3,74	**3,55**
An & Chung 2008 [31]	2,83	3,95	3,96	**3,58**
Choi & Kim 2008 [32]	2,82	4,04	3,92	**3,59**
Morency et al. 2008 [33]	2,91	4,97	3,67	**3,85**
Saragih et al. 2011 [34]	2,60	4,30	4,80	**3,90**
Tran et al. 2013 [35]	2,40	5,40	3,90	**3,90**
Tran et al. 2015 [36]	2,20	5,00	4,50	**3,90**
Mbouna et al. 2013 [37]	3,78	3,94	4,83	**4,18**
Cheung & Peng 2015 [38]	2,69	4,53	5,48	**4,23**
Vicente et al. 2015 [39]	3,20	4,30	6,20	**4,57**
Sung et al. 2008 [40]	3,10	5,40	5,60	**4,70**
Guo et al. 2012 [41]	5,30	4,90	4,80	**5,00**
Valenti et al. 2009 [42]	4,20	6,60	6,40	**5,73**
La Cascia et al. 2000 [24]	9,80	3,30	6,10	**6,40**

the templates in order to minimize error accumulation. Our basic approach follows in the ranking, with 2.85° in average. The rest of the methods from the literature obtain average errors above 3°, which shows to a greater extent the validity and accuracy of the methods presented in this thesis.

Translation results for the UPNA database are shown in Table 4.8. Translation estimation may be difficult to evaluate and results have to be analyzed carefully. The reason is that it has to be taken into account what spatial point in the object is the reference for which the position with respect to the camera is measured. Whereas rotation is independent from the origin at which the object coordinate system is located, translation is not. In the case of the UPNA database, the 3D ground truth is acquired with a sensor located on the top of the head of the user. However, the 3D head model reconstructed from the 2D observations through BA has the origin of the coordinate system approximately at the level of the nose, inside the head. This makes measuring absolute translation values unfeasible. Therefore, following the same procedure as for rotation evaluation, translation estimation and ground truth are aligned for the initial frame of each video sequence, and thus the differential error is measured. Table 4.8 shows that our complete method (wPOSIT+OD+OC) achieves a translation error of 11.08mm, significantly lower than the errors given by ASM or AAM combined with POSIT (22.40mm and 17.07mm respectively), and slightly

Table 4.8: HPE translation results for the UPNA database. Comparison between the four approaches proposed in this thesis and other reference methods. The differential HPE translation error is shown in all cases.

HPE Translation Results – UPNA Head Pose Database

METHOD	Error (mm)			
	Tx	Ty	Tz	Avg
Basic Approach	14,92	16,68	6,56	**12,72**
wPOSIT	13,54	15,52	6,08	**11,71**
wPOSIT+OD	12,81	15,60	6,18	**11,53**
wPOSIT+OD+OC	11,99	15,24	6,00	**11,08**
ASM + POSIT	26,51	27,61	13,09	**22,40**
AAM + POSIT	19,95	22,02	9,24	**17,07**
IF + POSIT	10,73	13,96	5,22	**9,97**

higher than the performance achieved by IF and POSIT (9.97mm in average). If we measure the gain given by each of our approaches with respect to the basic approach, we obtain 7.95%, 9.38% and 12.94% for wPOSIT, wPOSIT+OD and wPOSIT+OD+OC respectively for all the frames in the database, values that are comparable to those shown in the rotation error analysis.

4.6 Concluding Remarks

This chapter presents a global approach to the pose estimation problem, meaning that it is applicable to any kind of 3D object for which an approximate 3D model is available. Four different methods have been proposed: the basic approach, wPOSIT, wPOSIT+OD, and wPOSIT+OD+OC. Starting from the basic approach (2D tracking & POSIT), the weighted POSIT (wPOSIT) algorithm has been proposed, showing in a simulation environment that applying weights based on the current tracking accuracy of each point enhances the 3D pose estimation. Relying on this result, a method to calculate a tracking accuracy index (TAI) in real applications has been developed based on the proposal of two invariant shape metrics. The invariability is achieved by making the metrics independent from the specific shape configuration of the object (e.g. independent from the anatomical differences of different people), and by making them unaffected by object appearance variations in the image due to 3D pose changes. A tolerance model for the shape metrics has also been built in order to classify them as corresponding to a good or bad tracking, with which a TAI can be calculated and used to apply weights to POSIT. This method has been further developed to include an iterative outlier detection (OD) module based on thresholding, and an outlier correction (OC) module based on an accurate initial detection of 2D features and a pose denormalization process.

These global methods have then been adapted to the specific problem of HPE, including the implementation, analysis and discussion of different reference 2D facial feature detection and tracking algorithms necessary for the task. Two different databases of videos for HPE have been used for the evaluation of the proposed approaches: 1) the UPNA Head Pose Database, developed during this thesis and presented in Chapter 2, an extensive database of videos recorded with a commercial webcam in current technological conditions that, besides, provides a 2D ground truth for point tracking evaluation; 2) the BU Headtracking Database, recorded 15 years ago with poorer video quality and

noisier 3D ground truth, but with the advantage of being a widely referenced database for which many HPE articles in the literature have reported their results and thus allow an extensive performance comparison. Several conclusions can be drawn from this evaluation:

- Our best method achieves a 2D-tracking accuracy of 3.7 pixels for a 1280×720 video resolution and a 3D HPE accuracy of 1.13° for the UPNA database. These results represent the expected performance with a current commercial webcam.

- Each of the proposed approaches has proven to enhance the previous one, showing the validity of all the proposals. For the BU database, the performance with and without the OC module (wPOSIT+OD vs wPOSIT+OD+OC) is similar due to the poor quality of the videos. With current means, the OC module provides an extra accuracy gain and proves its value.

- The computational cost added by the proposed enhancements for the HPE system is low, and the methods presented in this work are well suited for real-time operation.

- The full approach (wPOSIT+OD+OC) achieves an accuracy gain, when compared with the basic approach, of almost 15% in normal tracking conditions, and over 25% in noisy tracking conditions.

- Our method clearly outperforms other methods of reference, widely accepted in the literature, such as ASM and AAM, both in 2D tracking and HPE when they are combined with POSIT. It also outperforms IF regarding the estimation of the 3D pose, and obtains comparable results to those given by the combination of IF's 2D tracking and POSIT (slightly worse in the UPNA database and slightly better in the BU database). IF is a very promising patent-pending face tracking method published in 2013, based on supervised learning; however, our method does not require any training and can besides be generalized for any 3D object tracking application.

- An extensive HPE accuracy comparison has been provided for the BU database, in which 19 works of reference in the literature have been included. Our method outperforms all of them, with an average error of 2.62°. Only two of the other 19 approaches obtain an accuracy below 3°, which shows to a greater extent the value and interest of the presented methods.

Bibliography of the Chapter

[1] D. F. DeMenthon and L. S. Davis, "Model-Based Object Pose in 25 Lines of Code," *Int. J. Comput. Vis.*, vol. 15, no. 1, pp. 123–141, 1995.

[2] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Proc. 7th Int'l Joint Conf. Artificial Intelligence*, 1981, vol. 130, pp. 674–679.

[3] X. Xiong and F. De Torre, "Supervised Descent Method and its Applications to Face Alignment," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 532–39.

[4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models-Their Training and Application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[6] K. Lekadir, R. Merrifield, and G. Yang, "Outlier Detection and Handling for Robust 3-D Active Shape Models Search," *IEEE Trans. Med. Imaging*, vol. 26, no. 2, pp. 212–222, 2007.

[7] I. Guttman, *Statistical Tolerance Regions: Classical and Bayesian*. London, U.K.: Griffin, 1970.

[8] A. Wald and J. Wolfowitz, "Tolerance Limits for a Normal Distribution," *Ann. Math. Stat.*, vol. 17, pp. 208–215, 1946.

[9] W. G. Howe, "Two-Sided Tolerance Limits for Normal Populations - Some Improvements," *J. Amer. Stat. Assoc.*, vol. 64, pp. 610–620, 1969.

[10] J. W. Pratt and J. D. Gibbons, *Concepts of Nonparametric Theory*. New York: Springer-Verlag, 1981.

[11] X. Xiong and F. De Torre, "Global Supervised Descent Method," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2664–2673.

[12] P. Viola and M. Jones, "Robust Real-Time Face Detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[13] T. F. Cootes and C. J. Taylor, "Active Shape Models - 'Smart Snakes,'" in *Proc. British Machine Vision Conference (BMVC)*, 1992, pp. 266–275.

[14] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," in *Proc. European Conference on Computer Vision (ICCV)*, 1998, vol. 2, pp. 484–498.

[15] T. Cootes, G. Wheeler, K. Walker, and C. Taylor, "Coupled-View Active Appearance Models.," in *British Machine Vision Conference*, 2000, vol. 1, pp. 52–61.

[16] I. Matthews and S. Baker, "Active Appearance Models Revisited," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 135–164, 2004.

[17] S. Baker, I. Matthews, and J. Schneider, "Automatic Construction of Active Appearance Models as an Image Coding Problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1380–1384, 2004.

[18] F. De Torre and M. H. Nguyen, "Parameterized Kernel Principal Component Analysis : Theory and Applications to Supervised and Unsupervised Image Alignment," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, 2008.

[19] J. Gonzalez-Mora, N. Guil, E. L. Zapata, and F. de la Torre, "Efficient Image Alignment using Linear Appearance Models," *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2230–2237, 2009.

[20] J. J. Cerrolaza, A. Villanueva, and R. Cabeza, "Hierarchical statistical shape models of multiobject anatomical structures: application to brain MRI.," *IEEE Trans. Med. Imaging*, vol. 31, no. 3, pp. 713–24, Mar. 2012.

[21] B. Amberg, A. Blake, and T. Vetter, "On compositional Image Alignment, with an application to Active Appearance Models," *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1714–1721, Jun. 2009.

[22] M. Ariz, J. J. Bengoechea, A. Villanueva, and R. Cabeza, "A Novel 2D/3D Database with Automatic Face Annotation for Head Tracking and Pose Estimation," *Comput. Vis. Image Underst.*, no. Special Issue on Assistive Computer Vision and Robotics, 2016 (in press).

[23] J. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature

Tracker Description of the algorithm," *Intel Corp. Microprocess. Res. Labs, OpenCV Doc.*, 1999.

[24] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: anapproach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, 2000.

[25] H. Wang, F. Davoine, V. Lepetit, C. Chaillou, and C. Pan, "3-D Head Tracking via Invariant Keypoint Learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 8, pp. 1113–1126, 2012.

[26] J. Xiao, T. Moriyama, T. Kanade, and J. F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *Proc. Fifth IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2002.

[27] S. Lefèvre and J. M. Odobez, "Structure and Appearance Features for Robust 3D Facial Actions Tracking," in *Proceedings IEEE International Conference on Multimedia and Expo, ICME*, 2009, pp. 298–301.

[28] B. H. P. Prasad and R. Aravind, "A Robust Head Pose Estimation System for Uncalibrated Monocular Videos," in *Proc. 7th Indian Conf. Computer Vision, Graphics and Image Processing*, 2010, pp. 162–169.

[29] J.-S. Jang and T. Kanade, "Robust 3D Head Tracking by Online Feature Registration," *IEEE Int. Conf. Autom. Face Gesture Recognit.*, pp. 1–6, 2008.

[30] S. Asteriadis, K. Karpouzis, and S. Kollias, "Visual Focus of Attention in Non-calibrated Environments using Gaze Estimation," *Int. J. Comput. Vis.*, vol. 107, no. 3, pp. 293–316, 2014.

[31] K. H. An and M. J. Chung, "3D Head Tracking and Pose-Robust 2D Texture Map-Based Face Recognition using a Simple Ellipsoid Model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2008, pp. 307–312.

[32] S. Choi and D. Kim, "Robust head tracking using 3D ellipsoidal head model in particle filter," *Pattern Recognit.*, vol. 41, no. 9, pp. 2901–2915, Sep. 2008.

[33] L. P. Morency, J. Whitehill, and J. Movellan, "Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation," in *IEEE International Conference on Automatic Face & Gesture Recognition*, 2008, pp. 1–8.

[34] J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable Model Fitting by Regularized Landmark Mean-Shift," *Int. J. Comput. Vis.*, vol. 91, no. 2, pp. 200–215, 2011.

[35] N. Tran, F. Ababsa, M. Charbit, and D. Petrovska-delacr, "3D Face Pose and Animation Tracking via Eigen-Decomposition based Bayesian Approach," *Adv. Vis. Comput. Lect. Notes Comput. Sci.*, vol. 8033, pp. 562–571, 2013.

[36] N.-T. Tran, F. Ababsa, and M. Charbit, "A Robust Framework for Tracking Simultaneously Rigid and Non-rigid Face using Synthesized Data," *Pattern Recognit. Lett.*, vol. 65, pp. 75–80, 2015.

[37] R. O. Mbouna, S. G. Kong, and M. G. Chun, "Visual Analysis of Eye State and Head Pose for Driver Alertness Monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1462–1469, 2013.

[38] Y. Cheung, S. Member, and Q. Peng, "Eye Gaze Tracking With a Web Camera in a Desktop Environment," *IEEE Trans. Human-Machine Syst.*, vol. 45, no. 4, pp. 419–430, 2015.

[39] F. Vicente, Z. Huang, X. Xiong, F. De Torre, W. Zhang, and D. Levi, "Driver Gaze Tracking and Eyes Off the Road Detection System," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2014–2027, 2015.

[40] T. Kanade, J. Sung, and D. Kim, "Pose Robust Face Tracking by Combining Active Appearance Models and Cylinder Head Models," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 260–274, Jan. 2008.

[41] W. Guo, I. Kotsia, and I. Patras, "Tensor Learning for Regression," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 816–827, 2012.

[42] R. Valenti, Z. Yucel, and T. Gevers, "Robustifying Eye Center Localization by Head Pose Cues," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2009, pp. 612–618.

CHAPTER 5

Conclusions and Future Work

This Chapter summarizes the most significant results and conclusions reported throughout this thesis. The aim of Section 5.1 is to provide a compilation of the main contributions to verify the fulfillment of the objectives proposed in Chapter 1. Furthermore, future research trends related to this work are proposed and briefly analyzed in Section 5.2.

5.1 Conclusions

Most of the work carried out in this Ph.D. thesis has been based on researching techniques to contribute to the development of accurate and robust head pose estimation (HPE) methods. This was motivated by the proliferation of human-computer interfaces (HCI) and the growing integration of HPE in this kind of systems, which builds another bridge for the interaction between humans and computers. In the following paragraphs, general conclusions drawn from this thesis are presented, following the structure of the document.

Different tools have been developed and presented in Chapter 2 in order to build an essential experimental framework for this thesis. It consists of a novel head pose database of videos of real subjects (UPNA Head Pose Database), and a complete simulation environment, including a synthetic head pose database of videos. The UPNA database provides a realistic experimental framework for

HPE, and its usefulness has been shown along the thesis: the 3D ground truth has been used for the assessment of HPE methods, the annotated 3D faces provide an ideal 3D model of each user for 3D face reconstruction evaluation, and the 2D ground truth landmarks have been used for face-tracking algorithm training and evaluation. The simulator tool, on the other hand, provides a totally controlled environment in which different sources of HPE error may be studied independently, and the synthetic database has become a very valuable tool for 3D face reconstruction experiments in 2D-tracking noise-controlled conditions. Furthermore, the simulation environment has made a thorough analysis of POSIT algorithm possible, showing its feasibility for the task of HPE.

The problem of 3D face reconstruction based on just 2D images has been addressed in Chapter 3. Various single-view and multiple-view based reconstruction methods have been compared over noisy and noise-free versions of the synthetic head pose database using two quantitative metrics: 1) the 3D fitting error, measured as point-to-point Euclidean distance from the reconstruction to the real face, and 2) the HPE error obtained by using the reconstructed face in the POSIT algorithm. This extensive study has led to, on the one hand, selecting the bundle adjustment (BA) method as the preferred reconstruction algorithm for our task and, on the other hand, drawing some interesting general conclusions: 1) there is an indetermination in the 3D model fitting process caused by the fact that the global scale of the head cannot be determined from just 2D observations; 2) multi-view approaches perform better than single-view ones, in great measure because determining the relative depths of the face with accuracy is critical for HPE; 3) as 2D tracking noise increases, its effect on the HPE error tends to prevail over the effect of the inaccuracies of the 3D face reconstruction; 4) in the presence of noisy 2D tracking, it is of critical importance for HPE performance to obtain an alignment between the estimation and the ground truth; in a real application, this turns into finding a corrective transformation for a noise-free (or low-noise) reference frame.

Finally, pose estimation has been addressed in Chapter 4 using a global approach, meaning that it is applicable to any object for which an approximate 3D model is available. We believe that developing methods that are generalizable to any kind of object gives added value to this Ph.D. thesis. Starting from a basic pose estimation approach (2D tracking & POSIT), different alternatives have been developed to improve performance. On the one hand, a tracking accuracy index (TAI) calculation method has been proposed, based on invariant shape metrics obtained from interlandmark relationships. This allows us to apply a

weighted version of POSIT (wPOSIT), which compensates for 2D tracking inaccuracies by optimizing the 3D pose estimation. On the other hand, outlier detection and outlier correction methods that aim to improve the 2D tracking itself have been proposed, addressing the typical *drifting* problem of point-tracking systems, and hence improving the 3D pose estimation further. These global methods have then been specifically adapted to HPE and evaluated using two head pose databases: the UPNA database, which reflects the expected performance in current technological conditions, and the BU database, a widely referenced older database that allows an extensive comparison with other state-of-the-art HPE methods. The evaluation has shown that, by implementing our method for head tracking, we may expect a 2D-tracking accuracy under 4 pixels (1280×720 video resolution) and a 3D HPE accuracy slightly over 1° with current commercial webcams. The proposed enhancements lead to an accuracy improvement of between 15% and 25% with respect to the basic approach, depending on the amount of tracking noise present in the video, and our method outperforms the rest of the state-of-the-art HPE algorithms when compared in accuracy over the BU database.

5.2 Future Work

While the objectives specified in the introductory chapter have been successfully met, the work developed in this thesis opens new lines of research in the short and mid-term. Various possibilities are described in the following lines.

One of the most important applications of HPE is its integration in HCI systems, and more specifically in gaze tracking devices. As described in the introductory chapter of this work, gaze tracking systems suffer in unconstrained environments because of their sensitivity to head motion, and HPE provides critical information when doing gaze estimation in low resolution images as well. Since this is currently one of the main lines of research of our group, it would be very interesting to carry through the integration of the HPE methods developed in this thesis in a low-resolution gaze estimation system. Moreover, it would be desirable to design a system with a bidirectional information channel, in the sense that the HPE and the iris detection and gaze estimation modules make use of mutual cues in order to improve their performance. HPE provides very useful information for iris position determination and the consequent gaze estimation, and at the same time the latter may provide cues that improve further the HPE results presented in this work.

The experiments performed regarding the 3D face reconstruction problem also open an interesting line of work. Many approaches have been studied for the reconstruction of a sparse 3D model, and promising results have been obtained for the reconstruction of the whole head from the sparse point cloud, although it has been observed that a higher number of points is required in order to achieve an accurate dense model. An interesting procedure would be to rely on a sparse set of points with known 2D-3D correspondences and build a dense point cloud of unknown correspondences around, making use of this dense cloud in order to perform a full-head 3D reconstruction and recover a complete model of the person in front of the camera. This seems a feasible implementation and is likely to achieve more accurate full-head reconstructions.

Finally, in terms of application, it would be interesting to study the integration of the presented HPE methods in mobile devices, such as smartphones, tablets, etc. New problems may arise due to differences in the imaging conditions, such as shorter distance from the head to the camera, lower resolution cameras, rapid background changes, or severe illumination variations among others. These new challenges would probably require an adaptation of the HPE methods, and the achieved accuracy would determine their range of application. Furthermore, since the presented tracking and pose estimation methods are global for any kind of 3D object, it would be interesting to develop new applications beyond HPE. For instance, interactive 3D reconstructions of tourist attractions, such as historical buildings, could be achieved from different 2D views acquired with the camera of a mobile device of a person visiting the place.