E.T.S. de Ingeniería Industrial,

Informática y de Telecomunicación

Optimización de sistemas de seguimiento y modelos 3D para *Head Pose Estimation*

Máster en Ingeniería Biomédica

Trabajo Fin de Máster

Autor: *Andoni Larumbe Bergera*

Tutores: *Rafael Cabeza Laguna*

Arantxa Villanueva Larre

Pamplona, 27 de Abril de 2017

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Resumen

En medicina, existen procedimientos en los que se requiere conocer detalladamente si la cabeza del paciente se ha movido y, si es así, cuanto lo ha hecho respecto a una posición inicial. Una forma no invasiva de resolver este problema es a través de la Visión Artificial, más concretamente, mediante la estimación de posición 3D; en este caso estimación de la posición de la cabeza (*Head Pose Estimation*, HPE). El HPE consiste en estimar la posición de la cabeza del paciente mediante puntos 2D obtenidos de imágenes en distintos instantes de tiempo, y un modelo 3D de la cabeza del paciente.

El objetivo principal de este proyecto es el de comparar, por un lado, distintos modelos de cabeza 3D y, por otro, múltiples sistemas de seguimiento facial. Estas comparaciones se realizan a fin de obtener la combinación que proporcione una mejor estimación de la posición de la cabeza de forma no invasiva.

Palabras clave: Seguimiento facial, Estimación de posición de la cabeza, Método de descenso supervisado, IntraFace, POSIT

Abstract

In medicine, there are several procedures in which it is necessary to know precisely if a patients' head has moved and, if that is the case, how much it has done so with regards to an initial position. A non-invasive way to solving this problem is through Computer Vision, more specifically, through a form of 3D pose estimation known as Head Pose Estimation (or HPE). HPE estimates the pose of the patient's head using both a set of 2D points obtained from images taken at different points in time and a 3D model of the patient's head.

The aim of this project is to compare different 3D head models, as well as multiple facial-tracking systems. These comparisons will allow us to determine which combination provides a better estimation of a head's pose in a non-invasive way.

Keywords: Face tracking, Head Pose Estimation, Supervised Descent Method, IntraFace, POSIT

Acrónimos

Acrónimo	Significado
2D	2 Dimensions
3D	3 Dimensions
3DMM	3D Morphable Model
AAM	Active Appearance Model
AFW	Annotated Facial Landmarks in the Wild
ASM	Active Shape Model
BFM	Basel Face Model
HOG	Histogram of Oriented Gradients
HPE	Head Pose Estimation
IBUG	Intelligent Behaviour Understanding Group
IF	IntraFace
LFPW	Labeled Face Parts in the Wild
POSIT	Pose from Orthography and Scaling by Iterations
SDM	Supervised Descent Method
SIFT	Scale Invariant Feature Transform
SSDM	Surrey Supervised Descent Method
SSDR	Surrey Supervised Descent Regressor
SFM	Surrey Face Model

Índice de Contenido

Índice de figuras

Capítulo 1
Introducción

El objetivo de este capítulo es el de presentar los intereses que motivaron la elección del tema del proyecto, el marco teórico sobre el que se desarrolla y los objetivos que se pretenden conseguir. Por último, se realiza un resumen de los contenidos que presenta cada capítulo a fin de proporcionar una visión global del proyecto.

1.1 Motivación

La tendencia actual en todos los procedimientos médicos es intentar realizarlos lo menos invasivos posible y ofrecer la máxima comodidad para el paciente. Muchas de las soluciones ideadas para conseguirlo, pasan por utilizar dispositivos que permitan capturar algún tipo de imagen de la que extraer información. Un claro ejemplo de ello son las técnicas más novedosas de detección tumoral (TAC, PET, SPECT y MR) y tratamiento oncológico (radioterapia de haz externo, radiocirugía estereotáxica). La inclusión de los dispositivos de captura de imagen en los procedimientos médicos ha hecho que la Visión Artificial (en inglés, *Computer Vision*), esté tomando cada vez más importancia en mundo de la salud.

La Visión Artificial es un subcampo de la Inteligencia Artificial cuyo objetivo es el de construir sistemas artificiales que permitan reproducir, o incluso superar, la capacidad que tiene el ser humano de, mediante la vista y el cerebro, comprender el mundo que le rodea. Así pues, aplicando, por ejemplo, geometría proyectiva, estadística, u optimización a una imagen, somos capaces de extraer propiedades como pueden ser la forma, la iluminación, o el color de los objetos presentes en ella [1].

El origen de la Visión Artificial se remonta a 1963; año en el que Lawrence Gilman Roberts publicó su tesis doctoral "*Machine Perception of Three-Dimensional Solids*" en la que explica el funcionamiento de un programa que permite generar un modelo 3D de distintos poliedros a partir de fotografías de los mismos.

En la Figura 1.1 se puede observar cómo, a partir de la fotografía de dos poliedros (Figura 1.1-A), el programa es capaz de generar un dibujo lineal (Figura 1.1-C), transformarlo en un modelo 3D y, finalmente, mostrarlo desde cualquier punto de vista (Figura 1.1-D).

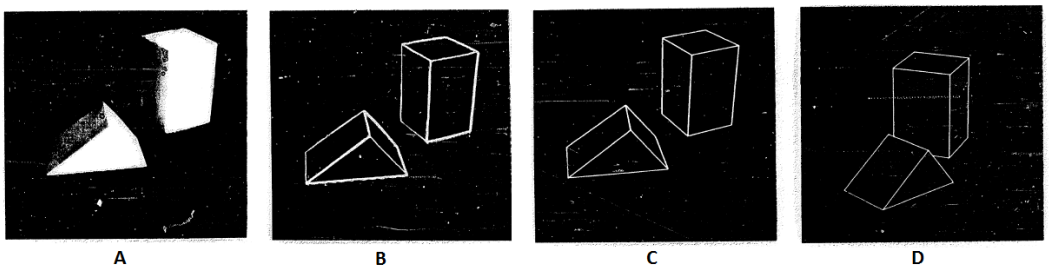

Figura 1.1 Rotación de solidos obtenidos a partir de una fotografía [2].

Desde entonces, la Visión Artificial se ha desarrollado considerablemente; hoy en día no se limita a simples polígonos sino que se utiliza en gran cantidad de aplicaciones de ámbitos muy distintos. Un pequeño ejemplo de las aplicaciones que existen hoy en día gracias a la Visión Artificial son:

- **Reconstrucción de modelos:** A partir de una serie de imágenes 2D de un objeto, se construye un modelo 3D del mismo. En la Figura 1.2 se muestra la reconstrucción 3D del coliseo a partir de más de 2000 imágenes realizadas por turistas y colgadas en la red social Flickr [3].

Figura 1.2 Reconstrucción del coliseo a partir de fotografías del mismo

- **Reconocimiento de objetos:** Detección de los distintos objetos, o incluso personas, que se encuentran en una imagen (ver Figura 1.3).

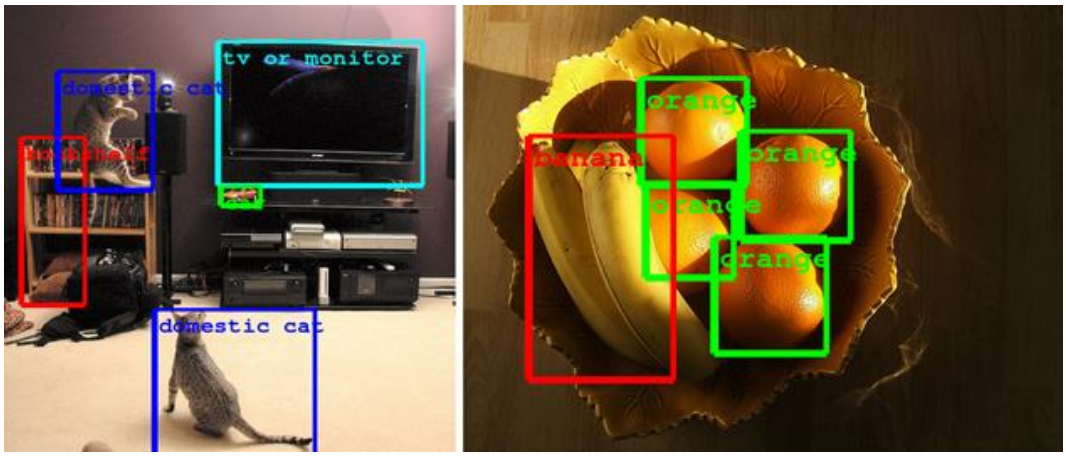

Figura 1.3 Reconocimiento de objetos en imágenes.

- **Alto rango dinámico:** A partir de imágenes de la misma escena captadas con diferente nivel de exposición (Figura 1.4, parte superior), se genera una nueva imagen que contenga un mayor rango dinámico (Figura 1.4, parte inferior).

Figura 1.4 High Dynamic Range Imaging.

Como se puede observar, la Visión Artificial puede servirnos de herramienta para resolver un problema, siempre y cuando dispongamos de las imágenes adecuadas.

Volviendo al tema de las técnicas de detección y tratamiento en el ámbito de la oncología, un problema existente en la oncología de cabeza y cuello es la presencia de movimientos de la cabeza por parte del paciente. Estos movimientos, por un lado, generan artefactos que disminuyen la calidad de imagen afectando la detección de posibles tumores (Figura 1.5, izquierda) y, por otro, aumentan el error de precisión de los tratamientos, lo que puede llegar a ser crítico según el tipo de tratamiento.

Un ejemplo de tratamiento oncológico en el que el movimiento del paciente es crítico es el *Gamma Knife*, un tipo de radioterapia en el que se aplica una única dosis de altísima radiación sobre un área pequeña de la cabeza. Durante el proceso, se utiliza un casco estereotáxico (Figura 1.5, derecha) que mantiene la cabeza inmóvil y garantiza que la dosis se focalice en el lugar exacto de la cabeza que necesita tratamiento.

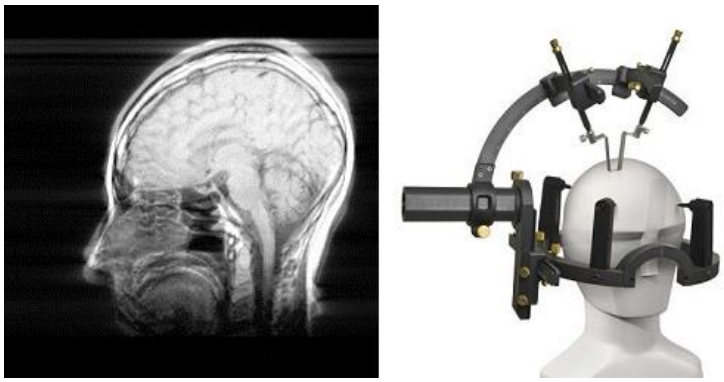

Figura 1.5 Consecuencias del movimiento de la cabeza.

Otro ejemplo de tratamiento, en este caso fuera del campo de la radioterapia, en el que el movimiento del paciente es crítico, es la estimulación magnética transcraneal; técnica en la que se estimula a una región concreta del cerebro mediante campos magnéticos. Un error en la localización de la zona a estimular puede tener consecuencias importantes.

El problema del movimiento del paciente en este tipo de procedimientos se puede abordar de dos formas muy distintas:

1. Utilizando sistemas de sujeción invasivos que impidan el movimiento del paciente; Ej. el casco estereotáxico.
2. Utilizando sistemas de captura de imagen y Visión Artificial para obtener la posición de la cabeza del paciente y emplearla para aplicar factores de corrección. La estimación de la posición de la cabeza se puede realizar mediante dos tipos de marcadores:

    - Marcadores fiduciales (Figura 1.6, izquierda).
    - Marcadores anatómicos calculados a partir de la propia imagen (Figura 1.6, derecha).

Figura 1.6 Tipos de marcadores utilizados para estimar la posición de la cabeza.

Resulta evidente que la forma más cómoda y menos invasiva de solucionarlo es mediante el uso de la Visión Artificial y los marcadores anatómicos. Esto se consigue mediante el denominado *Head Pose Estimation* (HPE), que explicaremos con detalle a lo largo del proyecto.

Otro tipo de aplicaciones del ámbito de la salud que hacen uso del HPE, son los denominados sistemas de estimación de la mirada, que buscan suplir las limitaciones de personas con movilidad reducida en tareas tan cotidianas como lo puede ser el manejo de un ordenador.

Una línea de investigación dentro del proyecto "*Interacción ubicua basada en la mirada para dispositivos móviles*" del grupo de Ingeniería Biomédica de la Universidad Pública de Navarra, trata de desarrollar un sistema de HPE con el objetivo de integrarlo en un sistema de estimación de la mirada, si bien como ya se ha comentado, es una aplicación que se puede aplicar en el campo que nos interesa. Es por tanto dentro de este grupo de investigación donde se ha decidido realizar el proyecto.

1.2 Estado del Arte

En el contexto de la Visión Artificial, el *Head Pose Estimation* es el proceso por el cual, mediante una serie de algoritmos, se consigue calcular la posición y la orientación de la cabeza de las personas que aparecen en una imagen. A lo largo del proyecto, se utilizará el término "pose" para describir al conjunto posición-orientación. La pose tiene 6 grados de libertad: tres ángulos de rotación (*roll, yaw, pitch*) que determinan la orientación y tres traslaciones (T_x, T_y, T_z) que determinan la posición [4]. En la Figura 1.7 se especifican las traslaciones y rotaciones que definen la pose de la cabeza.

Hoy en día es posible implementar el HPE de múltiples formas. E. Murphy-Chutorian y M. M. Trivedi en su artículo "*Head pose estimation in computer vision: A survey"* [5] realizaron una clasificación de los métodos de HPE según la técnica empleada. Este articulo ha servido de guía para analizar el estado del arte.

La clasificación realizada por Murphy y Trivedi se puede dividir en cuatro grandes grupos; tres de ellos divididos según la perspectiva desde la que se aborde el problema de la estimación de la pose; y un cuarto que engloba las combinaciones entre métodos.

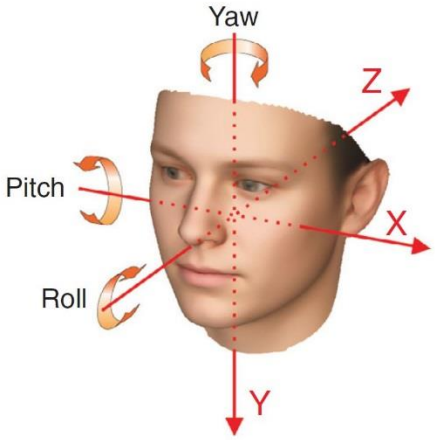

Figura 1.7 Sistema de coordenadas utilizado en el HPE [4].

1.2.1 Métodos basados en apariencia global

En el primer grupo, se encuentran las técnicas que intentan dar solución al problema de estimación de la pose extrayendo información del conjunto global de la imagen.

Los métodos que se pueden enmarcar dentro de este grupo son los patrones de apariencia, los arrays de detectores, la regresión no lineal y los métodos de manifold embedding.

1.2.1.1 Patrones de apariencia

Es la implementación más simple; consiste en comparar la imagen de la cabeza de una persona con un conjunto de imágenes que tienen asociada una pose concreta. La salida del sistema será la pose asociada a la imagen del conjunto que presente mayor similitud con la imagen de entrada (ver Figura 1.8).

Este tipo de implementaciones tienen algunas ventajas respecto a métodos más complejos. Una de ellas es que el conjunto de imágenes se puede modificar en cualquier momento; permitiendo expandirlo o adaptarlo a unas condiciones concretas. Además, añadir nuevas imágenes es algo sencillo ya que solo requiere asociarle una pose específica. Otra ventaja es que funciona para imágenes tanto de alta como de baja resolución.

Sin embargo, existen numerosas desventajas en el uso de estos métodos como por ejemplo que sin el uso de algoritmos de interpolación solo es capaz de estimar orientaciones discretas, lo que supone errores de estimación de alrededor de 10°. Pero sin duda, el mayor problema reside en que operan bajo la suposición de que la similitud en las imágenes está relacionada con la similitud en

pose; la imagen de un usuario en una pose determinada, tendrá mayor similitud con una imagen del mismo usuario en otra pose, que con una imagen de otro usuario en su misma pose. Para reducir este problema se han desarrollado numerosas soluciones, algunas de ellas se fundamentan en realizar algún tipo de procesado a la imagen.

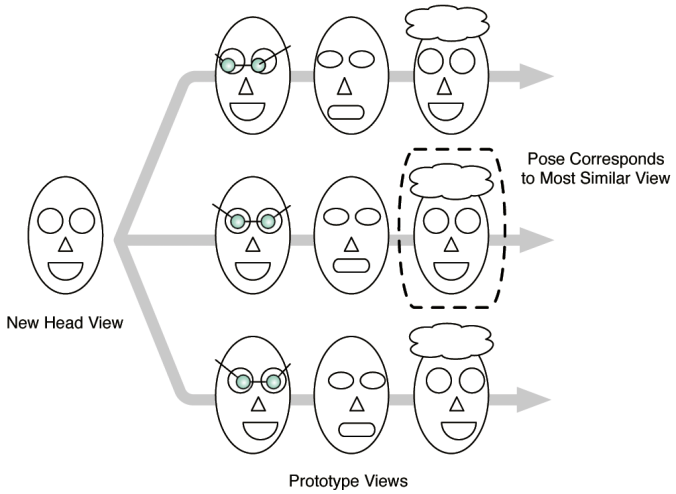

Figura 1.8 Esquema de funcionamiento del patrón de apariencia [5].

1.2.1.2 Array de detectores

Esta implementación está basada en los algoritmos de detección facial. La imagen de entrada es estudiada por múltiples detectores faciales, cada uno de ellos entrenado para detectar caras con una pose específica (ver Figura 1.9). Estos detectores pueden ser de dos tipos:

1. **Binarios:** cada detector proporciona únicamente "si" o "no" como salida en función de si ha detectado o no una cara. La pose de salida será la del detector que proporcione un "si". Se asume que no existen discrepancias entre detectores.
2. **Continuos:** cada detector proporciona un valor de confianza. La pose de salida será la asociada al detector con mayor valor.

La ventaja principal de estos sistemas reside en que no es necesario separar la detección facial de la estimación de la pose; es la propia detección facial la que nos proporciona información sobre la pose. Además, al emplear algoritmos de entrenamiento para los detectores faciales, soluciona el problema de que la similitud en las imágenes esté relacionada con la similitud en la pose.

Las desventajas están relacionadas con la carga computacional de entrenar detectores para un gran rango de orientaciones y con la dificultad de entrenar dos detectores de orientaciones similares que no se equivoquen. En la práctica (año 2009), estos sistemas están limitados a un único grado de libertad y menos de 12 detectores.

Al igual que en los métodos basados en patrones de apariencia, se consiguen errores de estimación de alrededor de 10°.

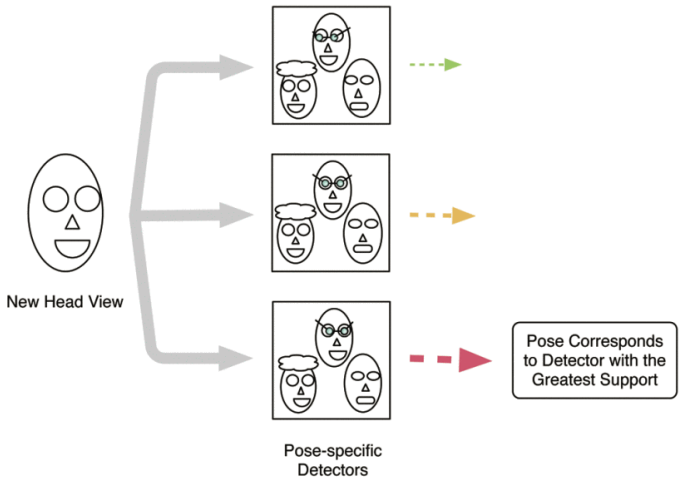

Figura 1.9 Esquema de funcionamiento del array de detectores [5].

1.2.1.3 Regresión no lineal

Este tipo de implementaciones intenta encontrar un mapeo funcional entre características propias de la imagen y la pose de la cabeza (ver Figura 1.10). Lo interesante de estos sistemas es su capacidad de, a partir de una serie de imágenes de entrenamiento, construir un modelo capaz de estimar la pose de nuevas imágenes. Para conseguirlo se pueden utilizar distintas herramientas de regresión, pero las más utilizadas son las redes neuronales.

Ejemplos de implementaciones que utilizan redes neuronales como herramienta de regresión son: el perceptrón multicapa (*multilayer perceptrón*, MLP), el cual puede ser entrenado para estimar orientaciones en un rango discreto o continuo; o el locally linear map (LLM), al cual se le puede aplicar descomposición por wavelets para reducir la dimensionalidad.

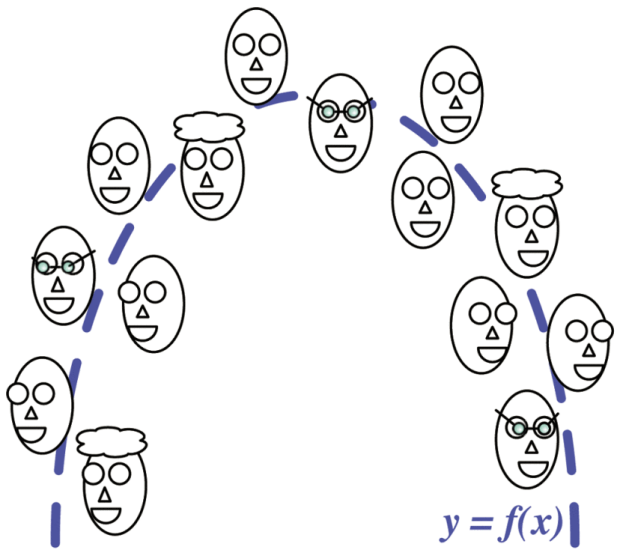

Figura 1.10 Esquema de funcionamiento de la regresión no lineal [5].

Son sistemas muy rápidos que proporcionan mejores resultados que los sistemas basados en apariencia (se consiguen errores de estimación de 5-10°) y que solo requieren imágenes etiquetadas con la pose de la cabeza. Sin embargo, su principal limitación es que no suelen proporcionar buenos resultados cuando no se tiene una buena localización de la cabeza.

1.2.1.4 Manifold embedding

Estas técnicas consisten en reducir la alta dimensionalidad del espacio imagen a las seis dimensiones de la pose. Dos de las técnicas más comunes para reducir la dimensionalidad son el análisis de componentes principales (PCA) y su versión no lineal kernelizada (KPCA).

El problema está en que al ser métodos no supervisados no hay ninguna garantía de que los primeros modos de variación dependan de la posición de la cabeza y no de otros factores como la apariencia o la iluminación.

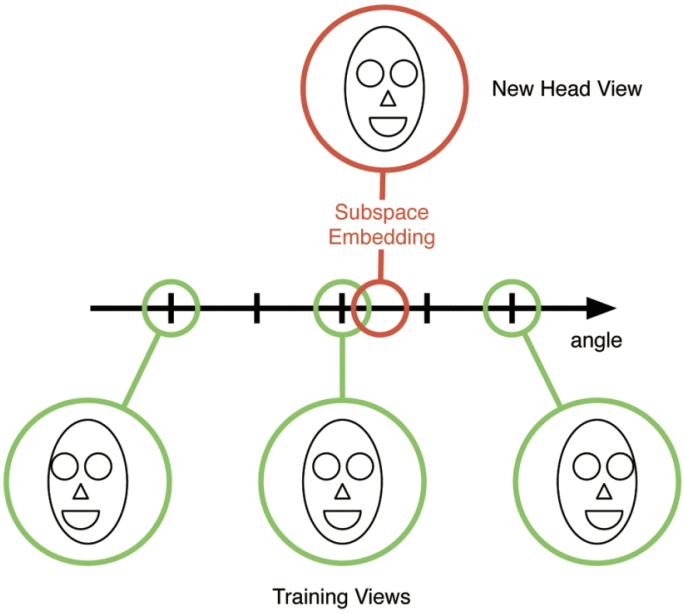

Figura 1.11 Esquema de funcionamiento del manifold embedding.

1.2.2 Métodos basados en apariencias locales

Los métodos incluidos en este grupo intentan dar solución al problema extrayendo información de características faciales locales (las esquinas de los ojos, la boca, la mandíbula, etc.; ver Figura 1.6, derecha). A lo largo del proyecto, se utilizará el término "landmark" para referirse a estas características faciales.

Los métodos definidos por E. Murphy-Chutorian y M. M. Trivedi que entrarían dentro de este grupo son los modelos deformables y los métodos geométricos.

1.2.2.1 Modelos deformables

Se intenta ajustar un modelo no rígido (deformable) de cabeza definido por una serie de landmarks a la estructura facial de la imagen. El resultado de ese ajuste proporciona información de la pose de la cabeza (ver Figura 1.12).

Las técnicas más utilizadas son: *Active Shape Model* (ASM) y *Active Appearance Model* (AAM). En ambas técnicas es necesario entrenar previamente con imágenes que tengan marcados los landmarks que se utilizan para realizar el ajuste. ASM aprende el comportamiento de la forma facial en las distintas poses mientras que AAM, además de aprender el comportamiento de la forma facial, aprende las texturas de las zonas vecinas a los landmarks y proporciona resultados más robustos.

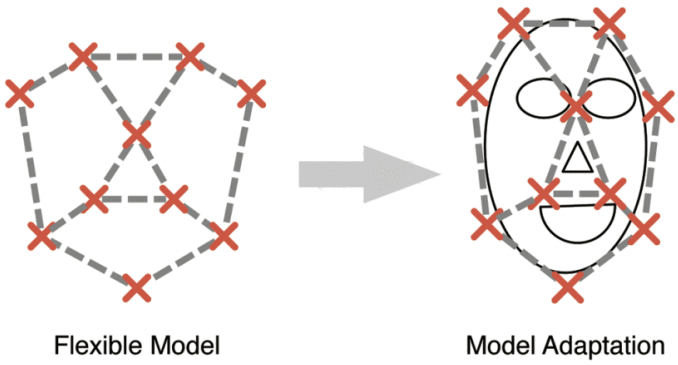

Figura 1.12 Esquema de funcionamiento del modelo deformable [5].

Al ser capaces de encontrar los landmarks para los que han sido entrenados, los AAM logran adaptarse a la imagen y ser muy precisos a la hora de estimar la pose facial (errores de estimación de 3-5°). Sin embargo, su mayor limitación es que están restringidos a imágenes con orientaciones faciales que permitan visualizar los landmarks con los que se haya entrenado (grandes valores de *yaw* impiden visualizar una parte de la cara)

1.2.2.2 Métodos geométricos

La pose se estima directamente de la configuración espacial de los landmarks. Las diferencias entre las distintas implementaciones de este tipo de métodos residen en los landmarks utilizados y las relaciones geométricas entre ellos (2D o 3D).

Por ejemplo, utilizando como landmarks las esquinas de los ojos, las esquinas de la boca y la punta de la nariz, se puede encontrar el eje de simetría a través de los puntos medios de los segmentos de los ojos y la boca (ver Figura 1.13). Una vez se tiene el eje de simetría, se puede calcular el ángulo 3D de la nariz y la pose facial.

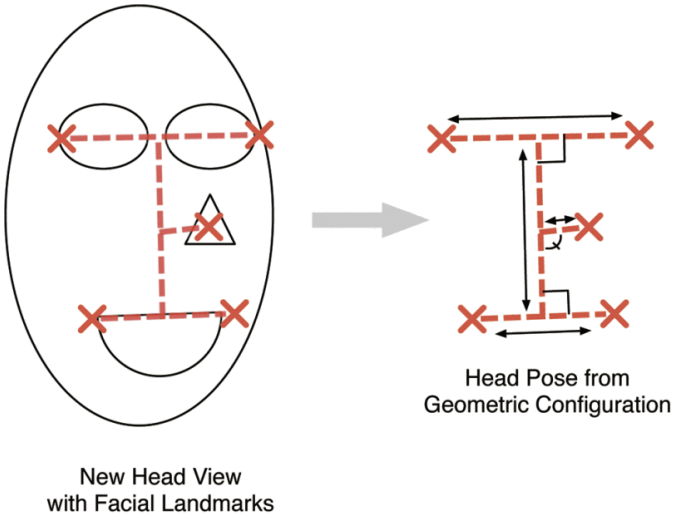

Figura 1.13 Esquema de funcionamiento del método geométricos [5].

Los métodos geométricos son rápidos, sencillos y proporcionan buenos resultados (3-5°). Utilizando un número no muy elevado de landmarks se mantienen resultados aceptables y se disminuye la carga computacional y el tiempo de procesado.

Su dificultad reside en detectar los landmarks de forma precisa. Por ejemplo, si las personas de las que se quiere estimar la pose utilizan gafas o si se tienen imágenes de baja resolución, la detección de los landmarks es más compleja y aumenta el error de estimación. Por ello, normalmente se utiliza una combinación de este tipo de técnicas con otro tipo de métodos para conseguir una estimación de la pose más robusta.

1.2.3 Métodos de seguimiento

Este tipo de métodos están enfocados a realizar un seguimiento del movimiento de la cabeza entre frames consecutivos (ver Figura 1.14). Su uso está bastante extendido porque, por lo general, existe un mayor interés en emplear HPE sobre vídeos y no sobre imágenes independientes.

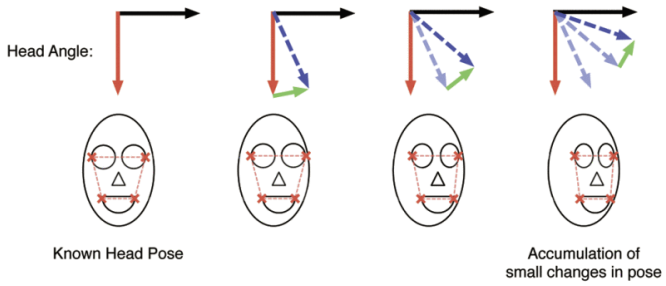

Figura 1.14 Esquema de funcionamiento del método de seguimiento [5].

Un tipo de implementación de este tipo de métodos se basa en utilizar modelos rígidos 3D para encontrar la transformación del modelo (rotación y traslación) que proporcione un mejor ajuste a los landmarks de cada frame. La separación de la localización de los landmarks y la estimación de la pose permite conocer la pose absoluta; sin esta separación únicamente se podría conocer la transformación respecto al frame anterior.

Estos sistemas presentan bajos niveles de error de estimación (1-3°) pero tienen el problema de necesitar una primera pose de inicialización. Normalmente, los sujetos tienen que mantener una pose frontal durante los primeros frames o se utilizan métodos de detección facial como el de Viola-Jones [6] para inicializar el seguimiento.

1.2.4 Métodos híbridos

En estos métodos se intenta dar solución al problema de estimación de la pose combinando distintas técnicas; de esta forma se intentan suplir las limitaciones que tiene cada una de ellas. En algunos casos esta combinación genera una nueva técnica, pero en otros, cada uno de los métodos realiza su propia estimación de forma independiente y, a partir de cada una de ellas, se calcula una estimación final.

Los métodos híbridos son los más utilizados en la actualidad debido a su versatilidad y a su capacidad de utilizar las ventajas de unas técnicas para compensar las limitaciones de otras. En la Figura 1.15 se ve un ejemplo de método hibrido en el que se combinan técnicas de apariencia global y de seguimiento.

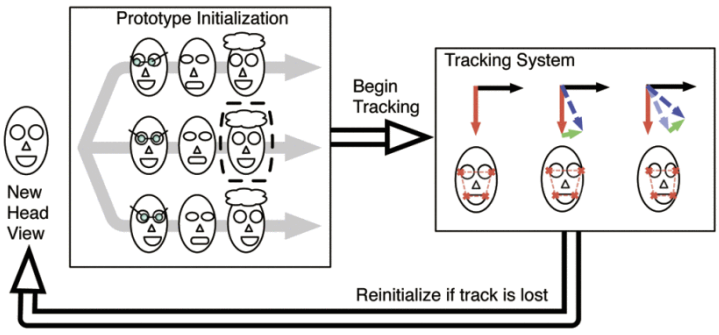

Figura 1.15 Esquema de funcionamiento del método híbridos [5].

Una de las combinaciones con las que se obtiene mejores resultados es la unión de un sistema de seguimiento de landmarks basado en el método de regresión SDM (*Supervised Descent Method*) [7]; y un método geométrico 3D. En este proyecto se ha optado por utilizar el método de estimación POSIT (*Pose from Orthography and Scaling by Iterations*) [8] que emplea esta combinación y presenta una gran eficiencia y precisión siempre y cuando los landmarks estén localizados correctamente.

Utilizando la localización de los landmarks en la imagen y un modelo 3D del objeto en el que se tenga la configuración geométrica 3D de dichos landmarks (ver Figura 1.16), y conociendo los parámetros de calibración de la cámara, POSIT permite estimar la pose del objeto 3D (Figura 1.17).

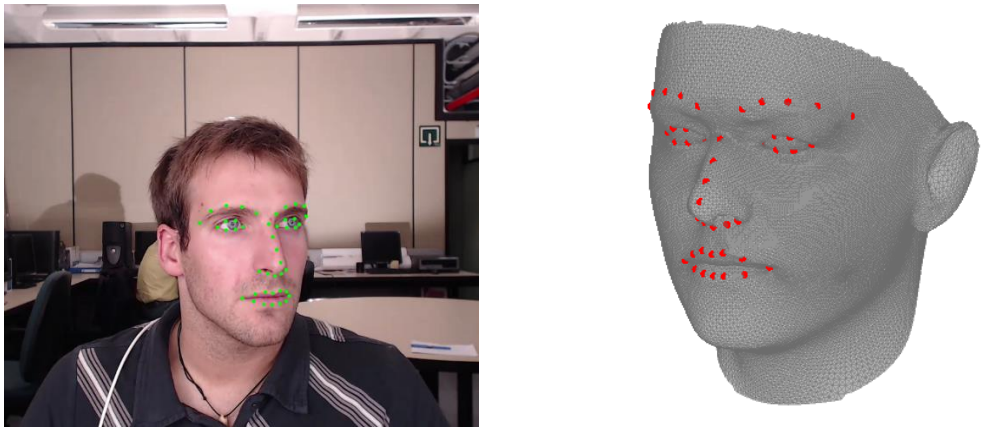

Figura 1.16 Localización de los landmarks en la imagen (izquierda) y modelo geométrico 3D (derecha).

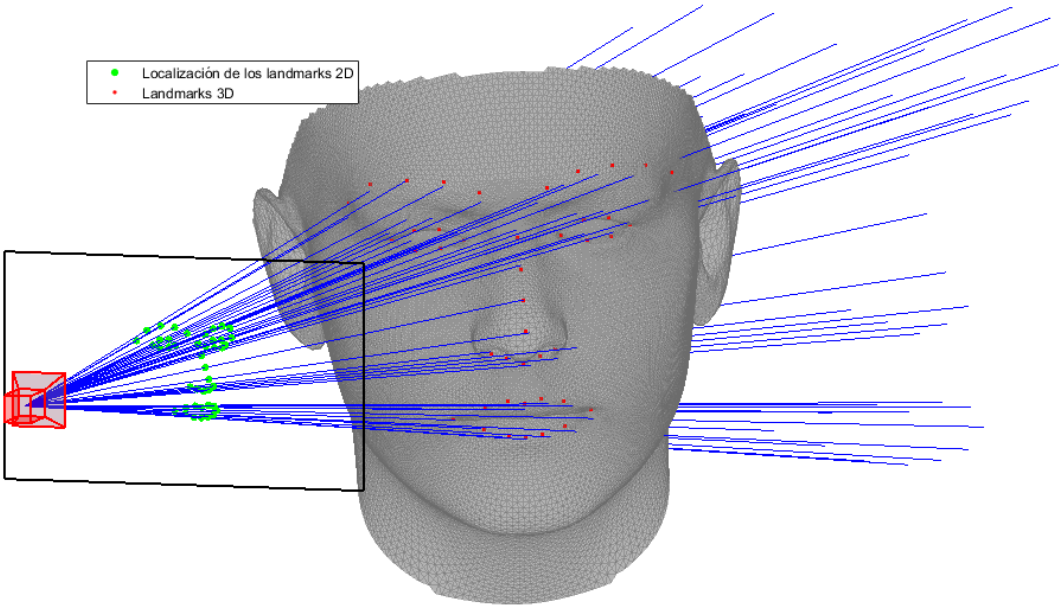

Figura 1.17 Estimación de la pose a partir de la localización de los landmarks en la imagen y en el modelo geométrico 3D.

1.3 Objetivos

El objetivo principal de este proyecto es obtener la combinación de modelo de cabeza 3D y sistema de seguimiento facial que proporcione la mejor estimación de la posición de la cabeza de forma no invasiva. Para conseguirlo, se deberá:

- Comparar diferentes modelos de cabeza 3D
- Evaluar si el uso de modelos 3D individualizados disminuye significativamente el error de estimación y, por consiguiente, merece la pena el cálculo de los mismos.
- Comparar y optimizar sistemas de seguimiento facial basados en métodos de descenso supervisado (*Supervised Descent Method*, SDM)

Como consecuencia de estos objetivos se ha desarrollado un entorno de trabajo con el cual poder comparar el HPE obtenido utilizando distintos sistemas de seguimiento y modelos 3D.

1.4 Estructura de la memoria

En el Capítulo 2 se presentan las herramientas utilizadas durante la realización del proyecto. Se detallan las bases de datos utilizadas así como los algoritmos empleados. También se explica con mayor profundidad el algoritmo POSIT, el algoritmo de regresión utilizado para el cálculo de los landmarks en la imagen y los modelos geométricos 3D.

En el Capítulo 3 se comentan las modificaciones realizadas sobre los modelos 3D para unificar los sistemas de coordenadas. También se detalla el procedimiento de retroproyección que permite obtener la correspondencia entre los puntos de cada modelo y los landmarks detectados por los sistemas de seguimiento. En el apartado final se describe el proceso utilizado para generar nuevos modelos 3D adaptados a cada usuario en particular.

En el Capítulo 4 se exponen las limitaciones de los sistemas de seguimiento facial y las implementaciones que se han desarrollado para solventarlas.

En el Capítulo 5 se muestran los resultados del error de estimación de pose obtenidos utilizando los distintos sistemas de tracking y modelos de cabeza 3D, y aplicando las implementaciones descritas en los capítulos anteriores.

Por último, en el Capítulo 6 se expone el conjunto de conclusiones obtenidas mediante la realización de este trabajo. También se dedica un apartado a comentar las líneas futuras que pueden surgir como consecuencia del mismo.

Capítulo 2
Framework

El objetivo de este capítulo es presentar las herramientas utilizadas a lo largo del proyecto. En primer lugar, se detallan las bases de datos que han permitido calcular y comparar el error de HPE. A continuación, se explica con mayor profundidad el algoritmo POSIT introducido en el capítulo anterior, el método de regresión utilizado para el cálculo de los landmarks en la imagen y los modelos geométricos 3D.

2.1 Bases de datos

En el capítulo anterior se han explicado distintas formas de realizar el HPE y se han comentado los errores de estimación que presentaba cada una de ellas. Para poder calcular estos errores es necesario hacer uso de bases de datos que asocien a cada uno de los sujetos que aparecen en las imágenes que la componen, el valor *groundtruth* (real) de pose.

En este proyecto se ha hecho uso de dos bases de datos; ambas realizadas en la Universidad Pública de Navarra y, como se explicará más adelante, relacionadas entre sí.

2.1.1 UPNA Head Pose Database [4]

La base de datos está formada por un total de 120 vídeos; 10 usuarios (6 hombres y 4 mujeres) y 12 vídeos por usuario. Cada conjunto de 12 vídeos correspondiente a un usuario se subdivide en dos grupos: 6 vídeos de movimiento guiado y 6 vídeos de movimiento libre. Cada vídeo de movimiento guiado presenta variaciones en una de las 6 dimensiones de pose (un único grado de libertad), mientras que los vídeos de movimiento libre pueden presentar variaciones en distintas dimensiones (6 grados de libertad).

En todos los usuarios se mantiene la misma estructura:

Vídeo 01: Traslación en el eje X
Vídeo 02: Traslación en el eje Y
Vídeo 03: Traslación en el eje Z
Vídeo 04: Rotación sobre el eje Z (*Roll*)
Vídeo 05: Rotación sobre el eje Y (*Yaw*)
Vídeo 06: Rotación sobre el eje X (*Pitch*)
Vídeos 07-12: Traslación y rotación libre

Los vídeos están grabados con una cámara web *Logitech HD Pro C920* a una resolución de 1280x720 y 30 frames por segundo. En la Figura 2.1 se representan frames de diferentes usuarios en distintas poses.

Figura 2.1 Frames de la UPNA Head Pose Database.

Cada vídeo tiene una duración de 10 segundos (300 frames) y tiene asociados 3 ficheros groundtruth, uno con las proyecciones 2D de un conjunto de landmarks (no coinciden con los landmarks utilizados en este proyecto, por lo que no se ha hecho uso de este fichero) y los otros dos con los valores de pose de cabeza; traslación dada en milímetros y rotación en grados. La diferencia entre los dos ficheros de pose de cabeza es que uno contiene el valor absoluto de pose en cada frame y el otro la pose de cada frame respecto a la del primer fotograma, es decir, simula que en el primer frame el usuario se encuentra completamente frontal.

La forma de obtener la pose es mediante el sistema de seguimiento magnético en tiempo real "*Flock of Birds, the 3D Guidance trackSTAR*" de la casa *Ascension Technology Corporation*. Este dispositivo está formado por un transmisor que sigue la posición de 4 sensores magnéticos a una frecuencia de 240 Hz. Dos de estos sensores se utilizaron para crear la base de datos: uno unido al sujeto a través de una diadema durante toda la grabación y otro utilizado para marcar la posición 3D de los landmarks respecto al primer sensor antes de la grabación.

2.1.2 UPNA Synthetic Head Pose Database [9]

Como ya se ha dicho, las bases de datos están relacionadas entre sí. Esto es así porque la segunda base de datos es una versión sintética de la primera.

A partir de las poses guardadas por el sistema de seguimiento magnético, se han generado 120 nuevos vídeos en los que se reproducen los movimientos de los usuarios de la primera base de datos, esta vez con cabezas generadas sintéticamente mediante el modelo deformable 3D *Basel Face Model* (BFM) [10] del que se hablará más adelante.

En la Figura 2.2 se ve como, comparando los frames de ambas bases de datos (Figura 2.1), las cabezas sintéticas tienen proporciones similares a las reales y la mayor diferencia reside en la ausencia de fondo.

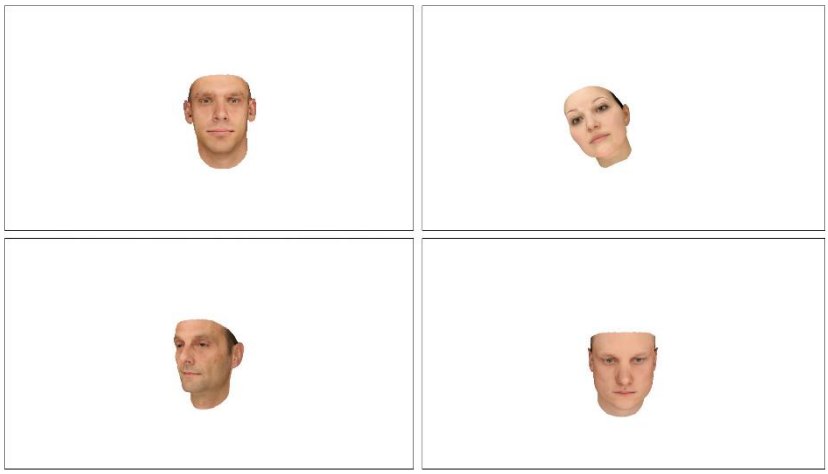

Figura 2.2 Frames de la UPNA Synthetic Head Pose Database [11].

Los vídeos sintéticos mantienen la resolución de 1280x720 y la tasa de 30 frames por segundo.

2.1.3 IBUG Facial point annotations

El *Intelligent Behaviour Understanding Group* (IBUG) es un grupo de investigación del *Imperial College London* centrado en el análisis del comportamiento humano a través de la computadora. Las áreas de aplicación de este grupo son: el análisis facial, el análisis de los gestos corporales, el análisis audiovisual del comportamiento humano, la biometría y *behaviometrics* y la interacción persona-computadora.

Dentro del área del análisis facial, uno de los campos de investigación son los sistemas de seguimiento de landmarks. Para su estudio, el grupo ha desarrollado una herramienta semiautomática de registro de landmarks con la que unificar diferentes bases de datos y poder así trabajar con un mayor número de imágenes [12]. Las bases de datos que han escogido son, entre otras: AFW, HELEN, LFPW y una base de datos propia de IBUG.

En la Figura 2.3, se muestra el conjunto de landmarks utilizado por el grupo IBUG con el que se han unificado las diferentes bases de datos. Se trata de un conjunto de 68 landmarks cuyo orden siempre es el mismo y están distribuidos entre mandíbula, cejas, ojos, nariz y boca.

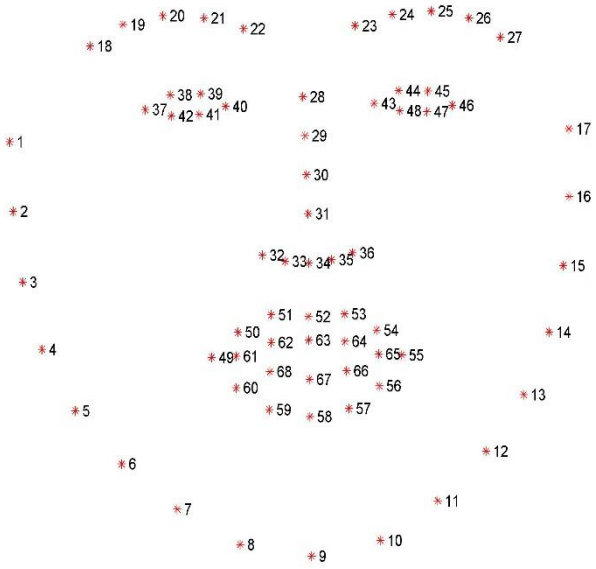

Figura 2.3 Conjunto de landmarks utilizado por IBUG.

2.1.3.1 AFW [13]

Annotated Facial Landmarks in the Wild (AFW o AFLW) es una base de datos creada en 2011 por el *Institute for Computer Graphics and Vision* de la Universidad de Graz (Austria). La base de datos original está formada por un total de 25.993 imágenes y define un conjunto de 21 landmarks (ver Figura 2.4).

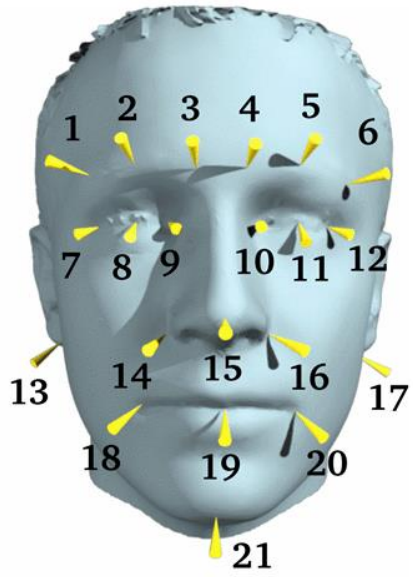

Figura 2.4 Landmarks definidos en AFW [13].

Por su parte, IBUG aplica su herramienta de registro sobre un subconjunto de 334 imágenes.

2.1.3.2 HELEN [14]

La base de datos HELEN está formada por un total de 2.330 imágenes y en ella se define un conjunto de 194 landmarks (ver Figura 2.5).

Figura 2.5 Imagen y landmarks de la base de datos HELEN.

En este caso, IBUG aplica su herramienta de registro al conjunto total de imágenes de la base de datos HELEN.

2.1.3.3 LFPW [15]

Labeled Face Parts in the Wild (LFPW) es una base de datos creada en 2011 y formada por un total de 1.432 imágenes obtenidas de páginas web como Flickr o Yahoo. Para cada imagen se define un conjunto de 29 landmarks (ver Figura 2.6).

Figura 2.6 Imagen y landmarks de la base de datos LPFW [15].

IBUG aplica su herramienta de registro sobre un subconjunto de 1.035 imágenes.

2.1.3.4 Base de datos propia de IBUG

El propio grupo IBUG ha generado una base de datos formada por un total de 135 imágenes sobre las cuales ha aplicado su herramienta de registro y obtenido los 68 landmarks (ver Figura 2.7).

Figura 2.7 Imagen y landmarks de la base de datos propia de IBUG.

2.2 POSIT [8]

Pose from Orthography and Scaling by Iterations es un algoritmo de estimación de pose publicado en 1995 por Daniel F. Dementhon y Larry S. Davis que, como ya se ha comentado, permite calcular la pose de un objeto a partir de una serie de landmarks en una imagen (landmarks 2D) y su correspondiente configuración geométrica en un modelo 3D (landmarks 3D).

POSIT está basado en el algoritmo POS (*Pose from Orthography and Scaling*). POS permite obtener el sistema de ecuaciones cuya solución es el valor aproximado de pose en la que el modelo 3D se proyecta sobre los landmarks 2D suponiendo perspectiva ortográfica escalada (ver Figura 2.8). La pose de salida está conformada por un vector de traslación $\boldsymbol{t}$ y una matriz de rotación **R** (a través de la cual se pueden calcular los ángulos *roll*, *yaw* y *pitch*).

Para que el algoritmo funcione es necesario un mínimo de cuatro puntos no coplanarios, aunque como cabe esperar, a mayor número de puntos, menor error de estimación [11]. Una de las ventajas de este algoritmo es que no es necesario realizar una inicialización de pose.

POSIT es un procedimiento iterativo que consta de tres pasos:

1. Se realiza POS sobre los landmarks 2D; como ya se ha dicho, suponiendo perspectiva ortográfica escalada. Si es la primera iteración, los puntos están definidos por la imagen de entrada (Figura 1.16, izquierda).

2. Se transforman los puntos que se han supuesto como perspectiva ortográfica escalada (P_i'), a las proyecciones perspectivas (a la misma profundidad) de la pose aproximada obtenida en el paso anterior (P_i).

3. Se comparan los puntos P_i obtenidos, con los de la iteración anterior. Si difieren significativamente, se repite el proceso utilizando como landmarks 2D los puntos P_i. Si los puntos no cambian respecto a la iteración anterior (o no lo hacen significativamente), la pose final es la pose aproximada en la última iteración.

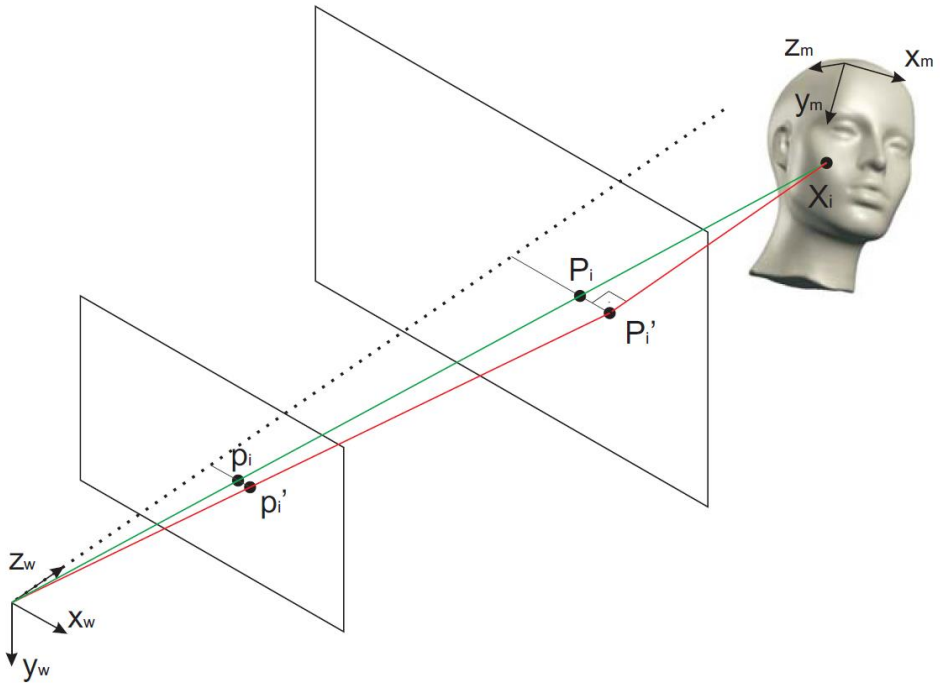

Figura 2.8 Diferencia entre perspectiva ortográfica escalada (rojo) y perspectiva proyectiva (verde).

2.3 Supervised Descent Method [7]

Muchos problemas de Visión Artificial se resuelven mediante el uso métodos de optimización no lineales. El método de descenso supervisado (SDM) es un algoritmo diseñado para resolver el problema de mínimos cuadrados no lineales (NLS) sin necesidad de utilizar matrices Jacobianas (**J**) ni Hessianas (**H**).

Esta característica es muy interesante porque, en el contexto de la Visión Artificial, el uso de este tipo de matrices tiene inconvenientes:

- La matriz Hessiana es definida positiva cerca de los mínimos locales, pero no se puede asegurar que lo sea en cualquier punto; la optimización puede no llegar a converger.
- Es necesario que las funciones se puedan derivar dos veces; gran cantidad de descriptores de características utilizados en Visión Artificial, como por ejemplo SIFT [16] o HoG [17], no son derivables.
- La matriz Hessiana tiene una alta dimensionalidad y operar con ella conlleva grandes tiempos computacionales.

En la Figura 2.9 se pueden ver las diferencias a la hora de minimizar la función $f(\mathbf{x}) = (h(\mathbf{x}) - y)^2$ entre el método de Newton (izquierda) y el SDM (derecha); h($\mathbf{x}$) es una función no lineal, $\mathbf{x}$ es el vector de parámetros a optimizar, e y es un escalar conocido. El eje z está invertido para una mejor visualización

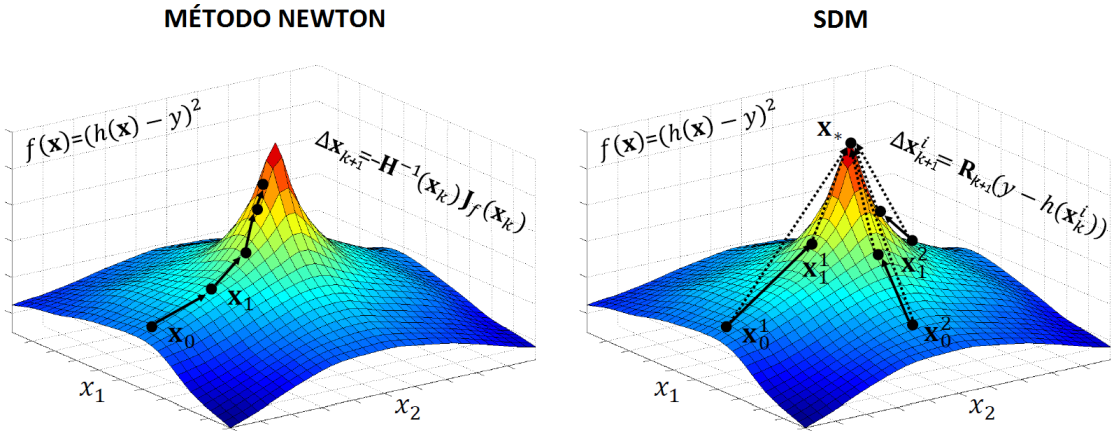

Figura 2.9 Comparación entre el método de optimización de Newton y el SDM [7].

La notación matemática utilizada en este trabajo es la siguiente:

- Letras mayúsculas en negrita hacen referencia a matrices: $\mathbf{X}$.
- Letras minúsculas en negrita hacen referencia a vectores columna: $\mathbf{x}$.
- Letras minúsculas sin negrita hacen referencia a escalares: x.
- $\mathbf{x_i}$ representa la i-ésima columna de la matriz $\mathbf{X}$.
- x_{ij} el escalar en la i-ésima fila y j-ésima columna de la matriz $\mathbf{X}$.
- x_j el escalar en la j-ésima columna del vector $\mathbf{x}$.
- $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x^T x}}$ denota la distancia Euclídea.

En el método de Newton, la secuencia de actualización se calcula de la forma:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}^{-1}(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) \quad \textbf{(2.1)}$$

Por lo que la trayectoria de optimización es:

$$\Delta\mathbf{x}_{k+1} = -\mathbf{H}^{-1}(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) \quad \textbf{(2.2)}$$

Otra forma de escribir las Ecuaciones 2.1 y 2.2, y que será la que se utilice en este trabajo, es la siguiente:

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{H}^{-1}(\mathbf{x}_{k-1})\mathbf{J}(\mathbf{x}_{k-1}) \quad \textbf{(2.3)}$$

$$\Delta\mathbf{x}_k = -\mathbf{H}^{-1}(\mathbf{x}_{k-1})\mathbf{J}(\mathbf{x}_{k-1}) \quad \textbf{(2.4)}$$

Se ve como, en cada iteración, el método de Newton requiere calcular los Jacobianos y los Hessianos, con todos los problemas que eso conlleva.

En el caso de SDM, es necesario realizar un entrenamiento previo. Durante el entrenamiento se muestrea una serie de puntos iniciales $\mathbf{x}_0^i$ alrededor del mínimo $\mathbf{x}_*$ (en el entrenamiento el mínimo es un valor conocido). Para cada una de estas muestras, la trayectoria de descenso ideal también se conoce (línea de puntos que une $\mathbf{x}_0^i$ con $\mathbf{x}_*$) y el algoritmo aprende la secuencia de actualizaciones que se aproximan a la trayectoria ideal $\mathbf{A}_k$ ($\mathbf{R}_k$ en la Figura 2.9).

Una vez el algoritmo ha sido entrenado para un caso particular, cada secuencia de actualización se calcula de la forma:

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{A}_k(\mathbf{h}(\mathbf{x}_{k-1}) - \mathbf{h}(\mathbf{x}_*)) \quad \textbf{(2.5)}$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{A}_k(\mathbf{h}(\mathbf{x}_{k-1}) - y) \quad \textbf{(2.6)}$$

La trayectoria de optimización queda:

$$\Delta\mathbf{x}_k = \mathbf{A}_k(y - \mathbf{h}(\mathbf{x}_{k-1})) \quad \textbf{(2.7)}$$

Como ya se ha dicho, SDM puede utilizarse para realizar el seguimiento facial mediante la localización de los landmarks 2D. Dada una imagen $\mathbf{d} \in \mathbb{R}^{mx1}$ con m pixels, $\mathbf{d}(\mathbf{x}) \in \mathbb{R}^{px1}$ hace referencia a los p landmarks y $\mathbf{h}$ es una función no lineal de extracción de características; por ejemplo $\mathbf{h}\big(\mathbf{d}(\mathbf{x})\big) \in \mathbb{R}^{128px1}$ en el caso de HoG.

El seguimiento facial puede implementarse minimizando la función:

$$f(\mathbf{x}) = \|\mathbf{h}(\boldsymbol{d}(\mathbf{x})) - \mathbf{y}_*\|_2^2 \quad \textbf{(2.8)}$$

Donde $\mathbf{y}_* = \mathbf{h}(\boldsymbol{d}(\mathbf{x}_*))$ son las características HoG extraídas de los landmarks conocidos. Sin embargo, existen problemas a la hora de realizar esta implementación; las caras sobre las que se va a aplicar el algoritmo no son las mismas que las utilizadas para entrenar al algoritmo, es decir, fuera del entrenamiento no conocemos $\mathbf{y}_*$; además, la función **h** no está parametrizada únicamente por los landmarks **x** sino que también depende de las imágenes (j). Para resolver estos problemas, durante el entrenamiento, el regresor SDM tiene que aprender un término de parcialidad $\mathbf{b}_k$ que represente la media de $\mathbf{A}_k\mathbf{h}^j(\mathbf{x}_*^j)$.

La trayectoria de optimización utilizando el término $\mathbf{b}_k$ quedaría:

$$\Delta\mathbf{x}_k = \mathbf{A}_k\mathbf{h}(\mathbf{d}(\mathbf{x}_{k-1})) + \mathbf{b}_k \quad \textbf{(2.9)}$$

Con esta modificación, SDM es capaz de detectar los landmarks 2D en imágenes de personas con las que no se ha entrenado o en presencia de cambios bruscos de iluminación.

Se puede definir como "regresor débil" al conjunto formado por la trayectoria de descenso y el término de parcialidad de cada iteración.

$$\mathbf{r}_k = \{\mathbf{A}_k\,, \mathbf{b}_k\} \quad \textbf{(2.10)}$$

El "regresor fuerte" será el regresor formado por K regresores débiles en serie; donde K es el número total de iteraciones.

$$\mathbf{R} = \mathbf{r}_1 \circ \mathbf{r}_2 \circ \cdots \circ \mathbf{r}_K \quad \textbf{(2.11)}$$

2.3.1 IntraFace [18]

IntraFace® (IF) es el software creado por los desarrolladores de la técnica del SDM por lo que la implementación del algoritmo es la descrita en el apartado anterior. En la Figura 2.10 se ve el conjunto de funcionalidades disponibles entre las que se encuentran el seguimiento facial y el HPE.

El problema de utilizar este software como sistema de seguimiento facial o de estimador de pose es que se trata de software de código cerrado; no hay forma de modificar su funcionamiento para adaptarlo a las necesidades de cada uno.

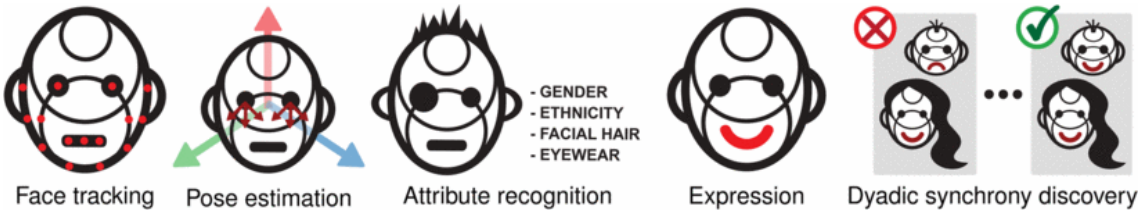

Figura 2.10 Funcionalidades de IntraFace.

En cuanto al regresor utilizando en IntraFace, el grupo no proporciona mucha información acerca de cómo ha sido entrenado aunque se sabe que utiliza un descriptor de características propietario denominado xx_SIFT (una modificación del descriptor SIFT).

En la Figura 2.11 se muestra el conjunto total de landmarks obtenidos por IntraFace, un total de 49. Comparándolo con los landmarks utilizados por IBUG (Figura 2.3) se observa que existe una clara analogía entre ambos conjuntos; la mayor diferencia reside en que IntraFace no utiliza los landmarks de la mandíbula (landmarks 1-17 utilizados en IBUG) ni los de las esquinas de los labios interiores (landmarks 61 y 65).

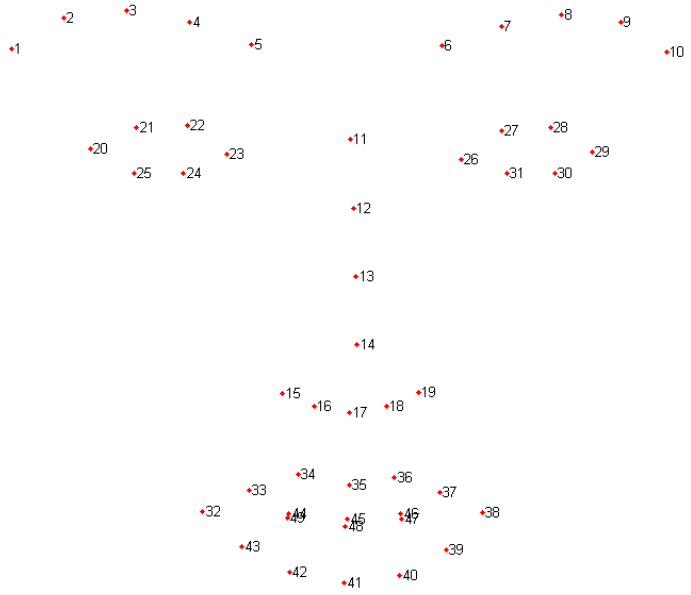

Figura 2.11 Conjunto de landmarks utilizado en IntraFace.

Tampoco se aportan detalles de cuál es la estrategia utilizada en cuanto a la inicialización de los landmarks (el conjunto inicial $\mathbf{x_0}$).

2.3.2 Surrey SDM

Surrey SDM (SSDM) es una implementación del algoritmo SDM realizada por la Universidad de Surrey y disponible en la plataforma online GitHub. En este caso se trata de software de código abierto (C++), por lo que es posible modificarlo y optimizarlo.

Las diferencias principales con IntraFace son:

- El algoritmo de detección de características: En este caso se utiliza una variante del descriptor HoG.
- La estructura del regresor: Utilizan el Random Cascaded-Regression Copse (R-CR-C); una estructura de regresor descrita por ellos mismos [19].
- Es una implementación invariante al tamaño de las caras.

La idea principal detrás del R-CR-C es utilizar una configuración de regresores en paralelo en vez de utilizar la configuración en serie definida en la Ecuación 2.11. El regresor está formado por un conjunto de "hilos" de regresores.

Se define la anchura W del regresor como el número de hilos de regresores que lo forman:

$$\mathbf{U} = \{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_W\} \tag{2.12}$$

La profundidad K es el número de regresores débiles (en serie) que constituyen cada uno de los hilos, por ejemplo para el regresor $\mathbf{R}_w$:

$$\mathbf{R}_w = \mathbf{r}_{w,1} \circ \mathbf{r}_{w,2} \circ \cdots \circ \mathbf{r}_{w,K} \tag{2.13}$$

En la Figura 2.12 se puede ver el esquema de entrenamiento de un R-CR-C con tres hilos y una profundidad D. Cada uno de los hilos se entrena con un subconjunto aleatorio de las imágenes de entrenamiento.

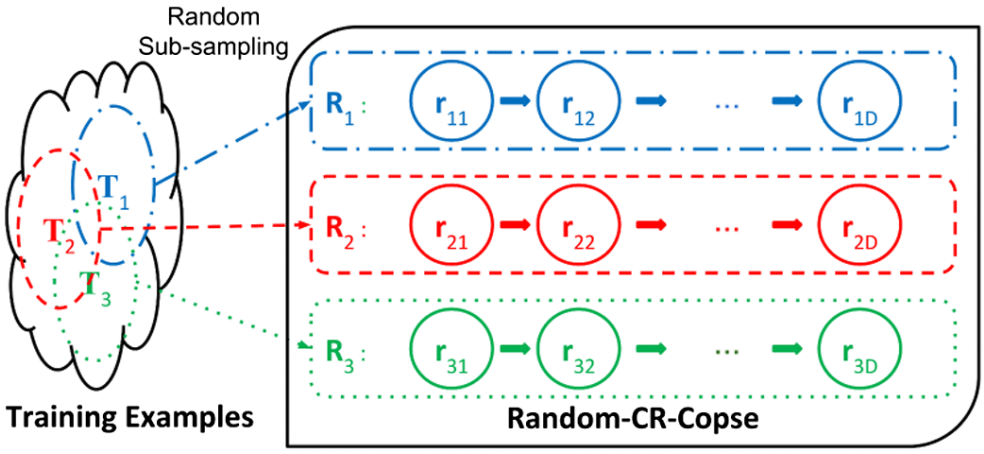

Figura 2.12 Esquema del Random Cascaded-Regression Copse [19].

En la detección se fusionan los resultados obtenidos por los distintos hilos, lo que proporciona un mejor balance entre el *over-fitting* y la pérdida de precisión.

Como ya se ha comentado, otra característica de esta implementación es que es robusta frente a la variación del tamaño de las caras. Esto es importante por dos motivos:

1. La extracción de características utilizadas para entrenar al algoritmo se aplica sobre el vecindario del valor groundtruth. En Figura 2.13 se puede observar cómo, a diferente escala, el vecindario del valor groundtruth, y por tanto las características extraídas de él, cambian completamente.

2. La diferencia entre el valor groundtruth y el valor inicial es totalmente dependiente de la escala.

Por ello, en el R-CR-C se define el parámetro s_f cuyo valor está relacionado con el tamaño de la cara. Este parámetro permite, para cada imagen, ajustar el tamaño del vecindario del que se extraen las características y entrenar al regresor para un tamaño de cara normalizado.

La trayectoria de optimización en la detección aplicando este factor de escala s_f es:

$$\Delta \mathbf{x}_{w,k} = s_f(\mathbf{x}_{w,k-1})\mathbf{A}_{w,k}\mathbf{h}\big(\mathbf{d}(\mathbf{x}_{w,k-1})\big) + \mathbf{b}_{w,k} \qquad \textbf{(2.14)}$$

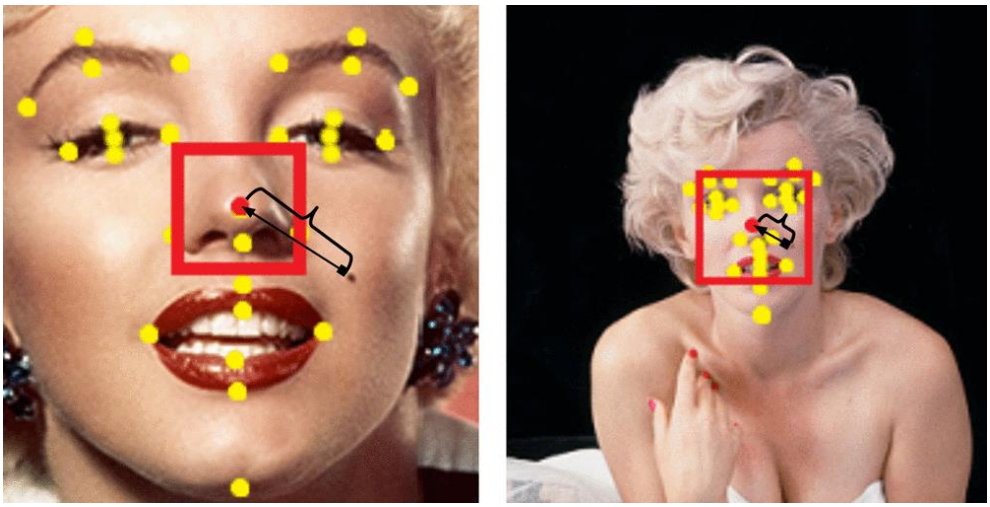

Figura 2.13 Consecuencias de la variación del tamaño de la cara [19].

SSDM incluye un regresor, el *Surrey Supervised Descent Regressor* (SSDR), así como el software necesario para poder entrenar nuevos regresores. El SSDR está entrenado con las bases de datos comentadas en el apartado 2.1.3 y los landmarks detectados por el software son los ilustrados en la Figura 2.3.

En cuanto al número de hilos de regresores que lo forman, la anchura W del regresor es de un total de 6 hilos.

La estrategia de inicialización utilizada por SSDM consiste en dos pasos:

1. Realizar la detección de la zona que encierra el conjunto de imagen en el que se encuentra la cara (el *bounding box*).

2. Fijar como conjunto inicial $\mathbf{x_0}$ las posiciones medias de los landmarks groundtruth respecto a sus bounding box. En este caso, las posiciones medias han sido calculadas utilizando la base de datos LPFW.

Existen distintos modos de generar el bounding box, uno de ellos es utilizar algoritmos de detección facial como el Viola-Jones [6]; otro es, si lo que se está realizando es un sistema de seguimiento, utilizar los landmarks detectados en el frame anterior (ver Figura 2.14, izquierda). En el segundo caso, el bounding box se define como el rectángulo cuyos vértices (ver Figura 2.14) están definidos por los valores x e y de:

1. Landmark con menor valor x y landmark con menor valor y.
2. Landmark con mayor valor x y landmark con menor valor y.
3. Landmark con mayor valor x y landmark con mayor valor y.
4. Landmark con menor valor x y landmark con mayor valor y.

Como ya se ha visto, en el caso de este proyecto, la detección de landmarks se realiza sobre bases de datos de vídeos por lo que el método con el que se generan los bounding box es utilizando los landmarks calculados en el frame anterior. Sin embargo, en el primer frame, al no tener landmarks previos, se realiza una detección facial utilizando el algoritmo Viola-Jones.

En la Figura 2.14 se puede observar un ejemplo del conjunto inicial $\mathbf{x_0}$ generado a partir del bounding box (derecha).

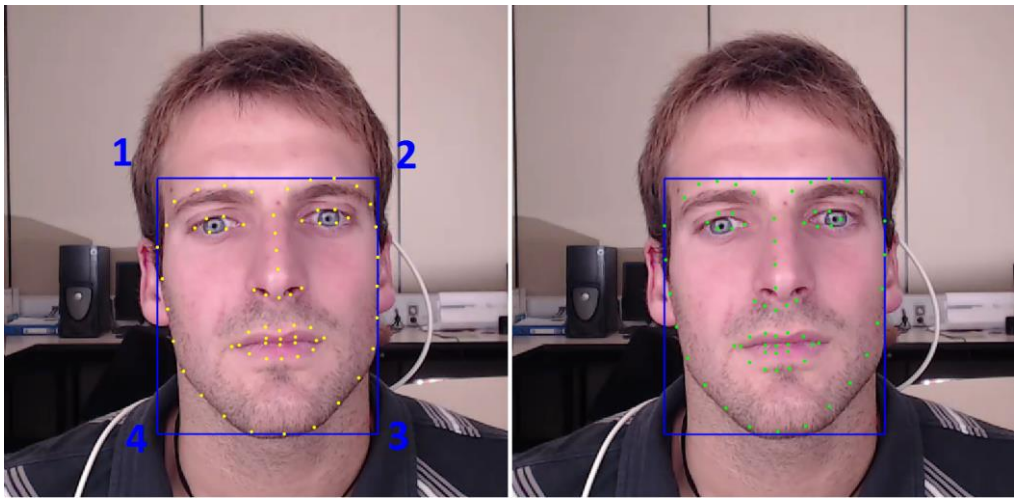

Figura 2.14 Bounding box generado a partir de los landmarks detectados en el frame anterior (izquierda) y conjunto inicial x_0 generado a partir del bounding box.

Al igual que IntraFace, SSDM también es capaz de realizar el HPE pero la diferencia reside en que, en este caso, no lo realiza utilizando SDM sino que utiliza un procedimiento geométrico basado en la correspondencia entre los landmarks 2D-3D [20].

2.4 Modelos 3D

Una vez vistos los sistemas de seguimiento de landmarks, el siguiente requisito para poder estimar la posición de la cabeza mediante POSIT son los modelos geométricos 3D. Los modelos geométricos 3D son nubes de puntos en el espacio de tres dimensiones (Figura 2.15, izquierda), estos puntos se encuentran asociados en tríos y conforman una malla triangulada (Figura 2.15, centro). Cada triangulo de la malla tiene asociado un color RGB que conforma la textura del modelo (Figura 2.15, derecha).

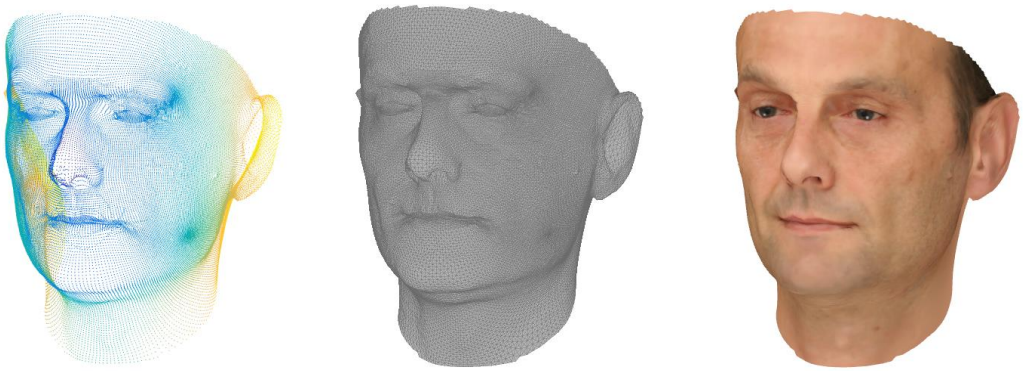

Figura 2.15 Modelo de cabeza 3D.

Mediante análisis de componentes principales (PCA) se pueden construir modelos deformables en los cuales, modificando los componentes principales, se consigan cambios en la forma y/o la textura del modelo (ver Figura 2.16).

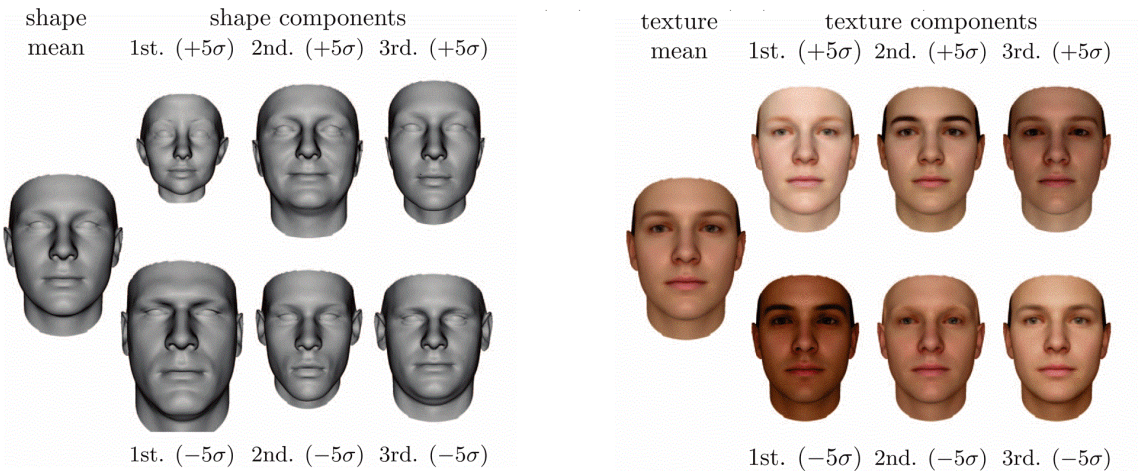

Figura 2.16 Variaciones de los componentes principales generan cambios en la forma o textura del modelo [10].

Los modelos 3D se utilizan para multitud de aplicaciones: estimación de pose, análisis facial, reconocimiento facial, seguimiento facial, detección de landmarks, etc. En este proyecto se hace uso de dos modelos deformables 3D: el Basel Face Model (BFM) y el Surrey Face Model (SFM).

2.4.1 Basel Face Model [10]

El BFM es un proyecto llevado a cabo por el *Computer Science Department* de la Universidad de Basel (Suiza) cuyo objetivo es el desarrollo de un modelo deformable 3D (ver Figura 2.17). Para generar el modelo, la Universidad de Basel realizó un escaneo 3D de 100 cabezas de hombres y 100 cabezas de mujeres de distintas edades.

Figura 2.17 Modelo medio BFM.

El mayor inconveniente con el uso de este modelo es que la Universidad de Basel no proporciona ningún tipo de software que ayude a manipular el modelo deformable y a generar modelos adaptados a cada usuario.

2.4.2 Surrey Face Model [20]

El SFM es un proyecto dentro del grupo *Centre for Vision, Speech And Signal Processing* de la Universidad de Surrey (Reino Unido). El número total de escaneos de cabeza fue de 169; entre los cuales había personas de distinta raza, edad y género.

Figura 2.18 Modelo medio SFM.

La Universidad de Surrey distribuye su modelo a distintas resoluciones (número total de puntos) y acompañado de archivos de metadatos (anotaciones de landmarks, mapas de textura…). En este caso, sí que se proporciona un entorno de trabajo con el que se facilita la manipulación del modelo.

Capítulo 3
Preparación de modelos

Uno de los requisitos para poder realizar el POSIT y, por tanto, poder calcular el error del HPE, es tener un modelo geométrico 3D en el que se conozca la correspondencia entre los puntos del modelo y los landmarks detectados por el sistema de seguimiento. Además, al utilizar modelos desarrollados por distintas entidades (BFM y SFM), los sistemas de coordenadas difieren entre sí y es conveniente unificarlos (ver Figura 3.1).

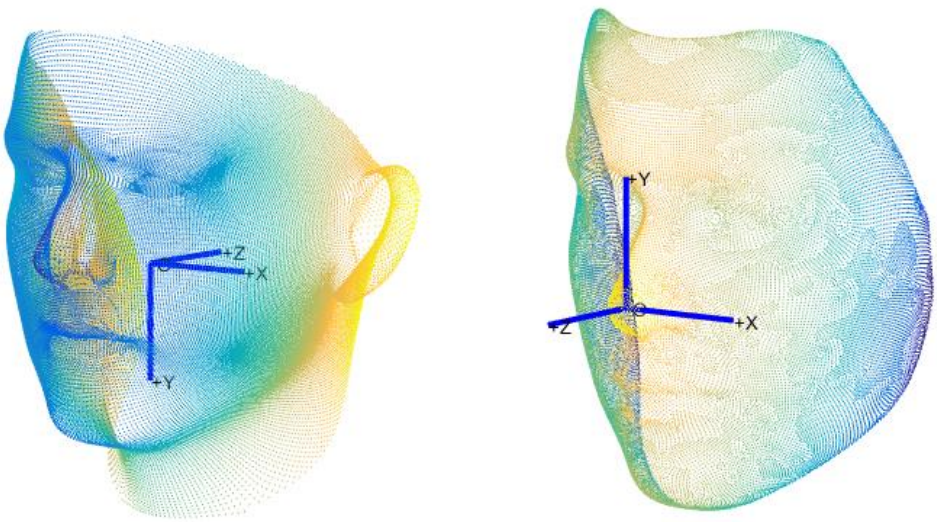

Figura 3.1 Sistema de coordenadas del modelo BFM (izquierda) y SFM (derecha).

En este capítulo se detalla el proceso seguido para unificar los sistemas de coordenadas de ambos modelos, así como los procedimientos que se han realizado para obtener la correspondencia entre los puntos de cada modelo y los landmarks detectados por los distintos sistemas de seguimiento.

Al final del capítulo se presenta una herramienta incluida dentro del entorno de trabajo del modelo SFM que permite, mediante modificaciones de los componentes principales del modelo, generar nuevos modelos 3D adaptados a cada usuario en particular.

3.1 Corrección del sistema de coordenadas

A la hora de unificar el sistema de coordenadas de los modelos geométricos es necesario definir tanto el origen como los ejes que conforman el sistema de coordenadas.

3.1.1 Ejes de coordenadas

Para unificar los ejes de coordenadas, se ha tomado como referencia el sistema utilizado en las cámaras digitales (ver Figura 3.2).

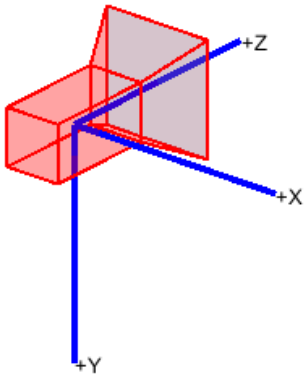

Figura 3.2 Sistema de coordenadas utilizado en las cámaras digitales.

Observando la Figura 3.1 se ve que el modelo BFM comparte los ejes de coordenadas con el sistema de las cámaras digitales, mientras que el SFM tiene los ejes Y y Z invertidos. La corrección de los ejes se realiza invirtiendo las coordenadas de los ejes Y y Z en el modelo SFM.

3.1.2 Origen del sistema

Para unificar el origen de coordenadas, se ha decidido utilizar como referencia el modelo BFM; de esta forma solo hay que corregir el origen del modelo SFM. En la Figura 3.1 se puede intuir que la diferencia en los orígenes de ambos modelos se puede corregir aplicando una traslación en el eje Z al modelo SFM.

En la Figura 3.3 se observa el resultado final tras aplicar la corrección del sistema de coordenadas al modelo SFM.

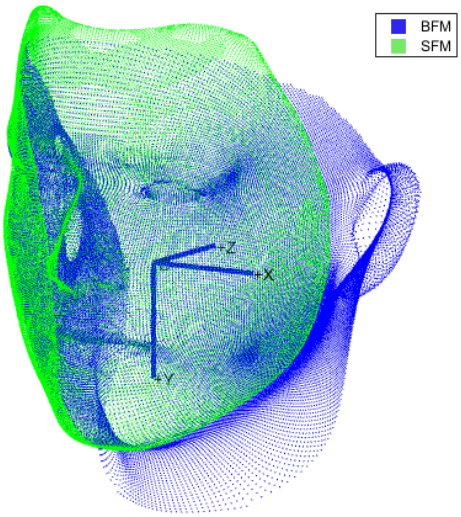

Figura 3.3 Modelos 3D corregidos

3.1.3 Sistema de coordenadas de la UPNA Head Pose Database

El sistema de coordenadas de las bases de datos de la UPNA está basado en el sensor magnético de la diadema que los usuarios llevan en la cabeza (ver Figura 3.4). Se puede observar que tanto el origen como la orientación de los ejes de coordenadas dependen de la posición del sensor; y ésta, varía en función del usuario (cada uno tiene la diadema puesta de forma distinta).

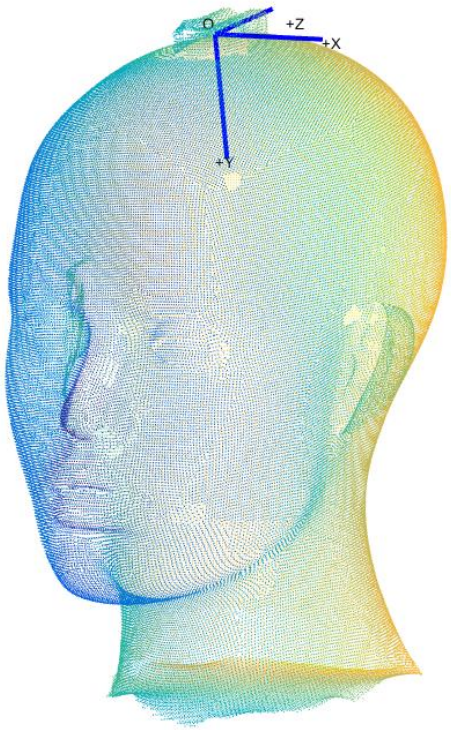

Figura 3.4 Sistema de coordenadas de las bases de datos de la UPNA.

Esto hace que, a la hora de calcular la pose y compararla con los ficheros groundtruth de la base de datos, existan diferencias que no son debidas a una mala estimación sino a las diferencias entre los sistemas de coordenadas. Para eliminar estas diferencias, la solución que se ha desarrollado se basa en referenciar la pose respecto al primer frame (hacer un *zeroed*) tanto en los valores groundtruth como en la estimación. De esta forma se consiguen obtener valores relativos a la posición inicial en los que los no importa, ni la diferencia en el origen, ni la pequeña rotación del sistema de coordenadas.

En los valores de pose estimados, la referencia se realiza guardando el valor estimado en el primer fotograma y restando este valor al valor estimado en todos los frames (en el primer frame los valores zeroed serán 0).

En cuanto a los valores groundtruth, en el apartado 2.1.1 ya se ha comentado que los vídeos de la base de datos de la UPNA tienen asociado un fichero en el que la pose de cada frame esta referenciada a la del primero.

3.2 Cálculo de landmarks 3D

Se denomina cálculo de landmarks 3D al proceso por el cual se obtiene la correspondencia entre los puntos del modelo 3D y los landmarks 2D detectados por el sistema de seguimiento. Como ya se ha comentado, los sistemas de seguimiento IntraFace y SSDM detectan landmarks similares (Figura 2.3 y Figura 2.11) pero no se puede asegurar que exista una relación directa entre ellos; por ello, el cálculo de landmarks 3D se deberá hacer para cada combinación de modelo 3D y sistema de seguimiento. Además, se ha decidido utilizar únicamente el conjunto de 43 landmarks representados en la Figura 3.5.

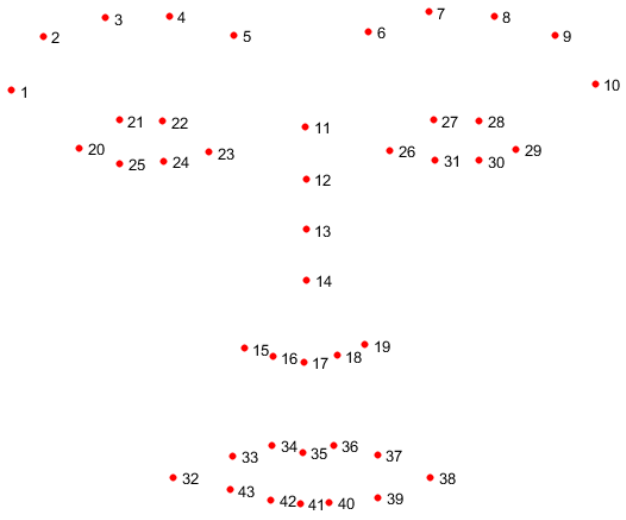

Figura 3.5 Conjunto de landmarks utilizado en este proyecto.

Las razones de utilizar este conjunto son que, por un lado, está definido tanto en IntraFace como en SSDM y, por otro, el resto de landmarks son propensos a generar errores debido a su localización en zonas limítrofes de la cara.

El cálculo de los landmarks 3D se realiza en dos pasos:

1. Generación de vídeos estáticos para obtener las posiciones medias de los landmarks 2D.
2. Retroproyección.

3.2.1 Generación de vídeos estáticos

Una característica de los sistemas de seguimiento es la presencia de *jitter*, es decir, pequeñas diferencias en la localización de los landmarks entre un frame y otro (incluso en ausencia de movimiento). Esto hace que, para realizar el cálculo de los landmarks 3D con mayor precisión, sea conveniente realizar la detección de los landmarks 2D múltiples veces sobre una misma imagen, y calcular la posición media de los landmarks 2D detectados.

La solución que se ha desarrollado para realizar la detección sobre una misma imagen múltiples veces, es generar sintéticamente vídeos del modelo 3D utilizando el software desarrollado para realizar la *UPNA Synthetic Head Pose Database*. Los vídeos consisten en un frame del modelo 3D en una posición frontal (ver Figura 3.6) repetido 100 veces.

Figura 3.6 Frame del modelo medio BFM.

3.2.2 Retroproyección

Una vez se tiene la media de los landmarks 2D, el siguiente paso es realizar el proceso denominado como retroproyección. La retroproyección es el cálculo de la intersección entre: la recta cuyo vector une el origen de coordenadas de la cámara y la posición del landmark medio en el plano imagen; y el plano formado por cada uno de los triángulos que conforman la malla del modelo 3D.

El algoritmo 1 describe el proceso por el cual se obtiene la correspondencia entre los landmarks 2D detectados por el sistema de seguimiento y los landmarks 3D.

Algoritmo 1 Retroproyección

1: Para cada landmark 2D **repito:**
2: Calculo el vector que une la posición del landmark 2D con el origen de la cámara
3: Normalizo el vector
4: Calculo la ecuación de la recta definida por éste vector y el origen de la cámara
5: Para cada triangulo que conforma la malla del modelo 3D **repito:**
6: Obtengo los vértices del triángulo
7: Calculo el vector normal del plano que contiene al triángulo
8: Calculo el punto de intersección recta-plano
9: **Compruebo:** Si la intersección se encuentra dentro del triángulo
10: Establezco el punto de intersección como landmark 3D.
11: **Fin de la comprobación**
12: **Fin de la repetición**
13: Fin de la repetición
14: Fin del algoritmo

Hay que tener en cuenta que el punto de intersección puede, y de hecho es lo más probable, no coincidir con ningún punto del modelo. Se ha desarrollado también una variación del algoritmo en la que, una vez calculado el punto de intersección, se calcule cuál de los vértices del triángulo está más cerca y se establezca a éste como landmark 3D (en este caso se le denominará vértice 3D). En la Figura 3.7 se puede ver la diferencia entre los landmarks 3D y los vértices 3D calculados a partir de los landmarks 2D detectados por el SSDR y el modelo SFM.

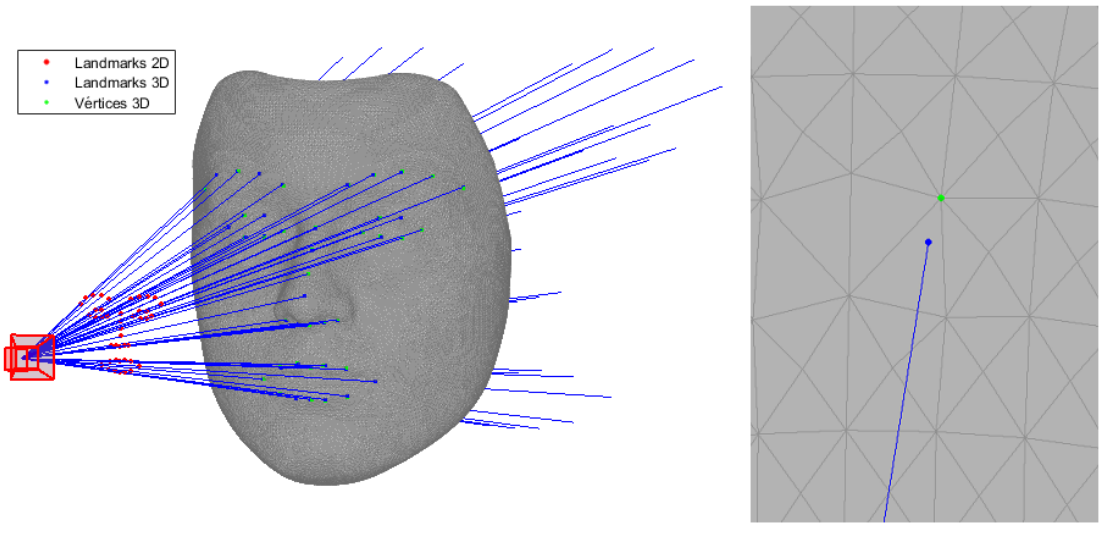

Figura 3.7 Diferencia entre landmarks 3D y vértices 3D.

3.3 Cálculo de modelos 3D específicos para cada usuario

Una de las herramientas proporcionadas por la Universidad de Surrey es el software *4DFace* [21], que permite, a partir del modelo SFM y de un conjunto de frames, deformar el modelo ajustándolo a usuarios particulares (ver Figura 3.8).

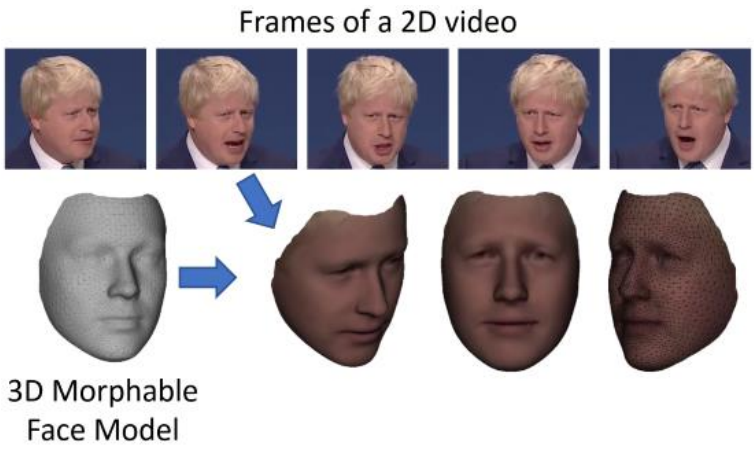

Figura 3.8 Esquema del programa 4DFace [21].

Utilizando este software se han calculado modelos particulares para los usuarios de las bases de datos real y sintética (ver Figura 3.9).

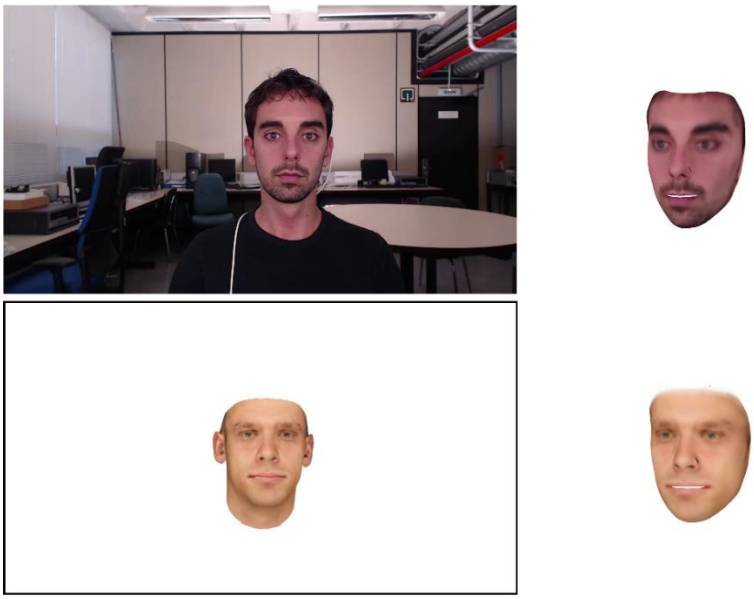

Figura 3.9 Ejemplo de generación de modelos particulares con la base de datos real (arriba) y sintética (abajo).

Hay que tener en cuenta que una vez tenemos los modelos reconstruidos para cada usuario hay que corregir su sistema de coordenadas. La corrección es igual que la aplicada al modelo SFM.

En este caso no se realiza retroproyección sino que se utilizan los vértices 3D calculados para el modelo SFM. Se ha tomado esta decisión porque la textura de los modelos reconstruidos no está perfectamente definida y al realizar la retroproyección no se obtienen resultados precisos.

Capítulo 4
Mejoras del tracking

Este capítulo está dedicado a explicar las limitaciones de los sistemas de seguimiento facial y las implementaciones que se han desarrollado para solventarlas. Como ya se ha dicho, IntraFace es un software de código cerrado y no se puede modificar su funcionamiento, por lo que todas las implementaciones detalladas en este capítulo están realizadas sobre el software SSDM.

4.1 Pérdida del tracker

Durante el proceso de localización de los landmarks, se observó que en algunos frames con valores altos de rotación (sobre todo en roll), los sistemas de seguimiento no daban buenos resultados. En la Figura 4.1 se puede ver el comportamiento del sistema SSDM (izquierda, marcas verdes) y de IntraFace (derecha, marcas amarillas) ante estos casos.

En el frame representado en la parte de arriba de la figura, el usuario se encuentra en una posición con un alto valor de roll. SSDM falla en la localización de landmarks, mientras que IntraFace mantiene una buena detección.

En el frame representado en la parte de abajo se observa que, mientras SSDM proporciona landmarks erróneos, IntraFace no proporciona ningún resultado. Esto es así porque IntraFace es capaz de detectar cuando se ha perdido para no proporcionar landmarks incorrectos.

Para corregir este error se ha decidido implementar dos soluciones que abordan el problema desde distinta perspectiva: la primera es entrenar al regresor con imágenes que presenten valores altos de rotación; la segunda es diseñar un algoritmo que calcule el valor de roll en cada frame y rote la imagen en consecuencia.

Como se verá más adelante, estas soluciones son perfectamente compatibles, por lo que la implementación final es una combinación de ambas dos.

Figura 4.1 Perdida de los sistemas de seguimiento SSDM (izquierda) e IF (derecha).

4.1.1 Entrenamiento de nuevos regresores

La primera solución consiste en generar regresores entrenados con un mayor número de imágenes o con imágenes que presenten mayores valores de rotación. Como ya se ha comentado, para realizar el entrenamiento es necesario el uso de imágenes con landmarks groundtruth. En nuestro caso utilizaremos las bases de datos empleadas para entrenar el SSDR y, además, la base de datos *UPNA Synthetic Head Pose Database*. Hay que tener en cuenta que una vez se entrenan nuevos regresores con frames de los vídeos sintéticos, ya no tiene sentido obtener resultados de error HPE sobre esta base de datos.

Una de las limitaciones al entrenar regresores es la capacidad finita de memoria RAM que tienen los ordenadores con los que se realiza dicho entrenamiento; al aumentar el número de imágenes con las que se pretende entrenar al regresor, también aumenta el requisito de memoria RAM. Esto hace que muchas veces, si no se tiene el equipo apropiado, no se puedan realizar los entrenamientos con la cantidad de imágenes que se desea.

4.1.1.1 Uso de la base de datos sintética en el entrenamiento

Para conseguir entrenar al regresor ante altos valores de rotación, ha sido necesario el uso de frames de la base de datos sintética. Por lo tanto, es necesario conseguir los landmarks groundtruth y el bounding box de cada frame empleado.

El proceso de obtención de los landmarks se realiza multiplicando los landmarks 3D obtenidos por retroproyección para cada modelo 3D, por una matriz de Rotación-Traslación (**RT**). Para construir la matriz **RT** se utilizan los ficheros de pose groundtruth de la base de datos.

Para cada frame se tienen los valores de traslación T_x, T_y y T_z por lo que se puede generar un vector de traslación **T** :

$$\mathbf{T} = [T_x \quad T_y \quad T_z] \tag{4.1}$$

También se tienen los valores de grados de rotación roll (γ), yaw (β) y pitch (α) por lo que se puede generar una matriz de rotación para cada dimensión.

$$\mathbf{R_{roll}}(\gamma) = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} CR & -SR & 0 \\ SR & CR & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

$$\mathbf{R_{yaw}}(\beta) = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} = \begin{bmatrix} CY & 0 & SY \\ 0 & 1 & 0 \\ -SY & 0 & CY \end{bmatrix} \tag{4.3}$$

$$\mathbf{R_{pitch}}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & CP & -SP \\ 0 & SP & CP \end{bmatrix} \tag{4.4}$$

Nótese que en las ecuaciones anteriores $CR = \cos(\gamma)$, $SR = \sin(\gamma)$, $CY = \cos(\beta)$ …

Multiplicando las tres matrices de rotación para una dimensión $\mathbf{R_{roll}}(\gamma)$, $\mathbf{R_{yaw}}(\beta)$ y $\mathbf{R_{pitch}}(\alpha)$, se obtiene la matriz de rotación en tres dimensiones $\mathbf{R}(\gamma, \beta, \alpha)$:

$$\mathbf{R}(\gamma, \beta, \alpha) = \mathbf{R_{roll}}(\gamma) \cdot \mathbf{R_{yaw}}(\beta) \cdot \mathbf{R_{pitch}}(\alpha) \tag{4.5}$$

$$\mathbf{R}(\gamma, \beta, \alpha) = \begin{bmatrix} CY \cdot CR & SP \cdot SY \cdot CR - CP \cdot SR & CP \cdot SY \cdot CR + SP \cdot SR \\ CY \cdot SR & SP \cdot SY \cdot SR + CP \cdot CR & CP \cdot SY \cdot SR - SP \cdot CR \\ -SY & SP \cdot CY & CP \cdot CY \end{bmatrix} \tag{4.6}$$

La matriz de Rotación-Traslación se define como:

$$\mathbf{RT} = \begin{bmatrix} \mathbf{R_{1,1}} & \mathbf{R_{1,2}} & \mathbf{R_{1,3}} & \mathbf{T_x} \\ \mathbf{R_{2,1}} & \mathbf{R_{2,2}} & \mathbf{R_{2,3}} & \mathbf{T_y} \\ \mathbf{R_{3,1}} & \mathbf{R_{3,2}} & \mathbf{R_{3,3}} & \mathbf{T_z} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{4.7}$$

Por último, y como ya se ha dicho, multiplicando esta matriz de rotación por cada landmark 3D (**L**), obtendremos la posición del landmark 3D rotado (**LR**):

$$\mathbf{L} = [\mathbf{x} \quad \mathbf{y} \quad \mathbf{z} \quad \mathbf{1}] \tag{4.8}$$

$$\mathbf{LR} = \mathbf{RT} \cdot \mathbf{L}^{\mathbf{T}} = \begin{bmatrix} \mathbf{R_{1,1}} & \mathbf{R_{1,2}} & \mathbf{R_{1,3}} & \mathbf{T_x} \\ \mathbf{R_{2,1}} & \mathbf{R_{2,2}} & \mathbf{R_{2,3}} & \mathbf{T_y} \\ \mathbf{R_{3,1}} & \mathbf{R_{3,2}} & \mathbf{R_{3,3}} & \mathbf{T_z} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{x_R} \\ \mathbf{y_R} \\ \mathbf{z_R} \\ \mathbf{1} \end{bmatrix} \tag{4.9}$$

En cuanto al bounding box, se define a partir de los landmarks groundtruth del frame anterior; de la forma ya vista en el apartado 2.3.2.

4.1.1.2 Inicialización de los landmarks

En primer lugar, se realizó una comparación de la eficacia del regresor al inicializarlo con los landmarks obtenidos en el frame anterior (en el caso de utilizarse con vídeos) o con los landmarks medios ajustados al bounding box (lo que hace SSDM).

Para ello se utilizó el software desarrollado por el doctor Zhenhua Feng (Universidad de Surrey) disponible en su perfil de GitHub. El software es una versión del algoritmo SSDM desarrollada en el lenguaje de programación MATLAB orientada a realizar seguimiento facial mediante el entrenamiento y uso de regresores. La razón de utilizar este software para realizar la comparación en vez de usar la implementación en C++, es precisamente que el entorno de MATLAB es más sencillo de utilizar.

El software original está pensado para utilizar los landmarks medios como inicialización del SDM, pero se ha realizado una modificación para poder utilizar los landmarks obtenidos en el frame anterior. Además, se ha realizado una serie de modificaciones en el algoritmo de entrenamiento que permiten aumentar el número de imágenes utilizadas, a costa de aumentar en gran medida los tiempos de cómputo.

Se han entrenado dos regresores, ambos con el mismo número de imágenes, pero con distinto tipo de inicialización de los landmarks. En el primero, los landmarks se inicializan con los valores obtenidos en el frame anterior mientras que en el segundo se inicializan con los landmarks medios ajustados al bounding box. En la Figura 4.2 se puede observar cómo, en el caso de utilizar el primer regresor, conforme pasan los frames, los landmarks se dispersan (arriba, landmarks rojos) mientras que usando el segundo regresor, el tracker funciona correctamente (abajo, landmarks magentas). Parece ser que el hecho de inicializarlo con los landmarks medios, actúa de alguna forma como regulador e impide la dispersión de los landmarks.

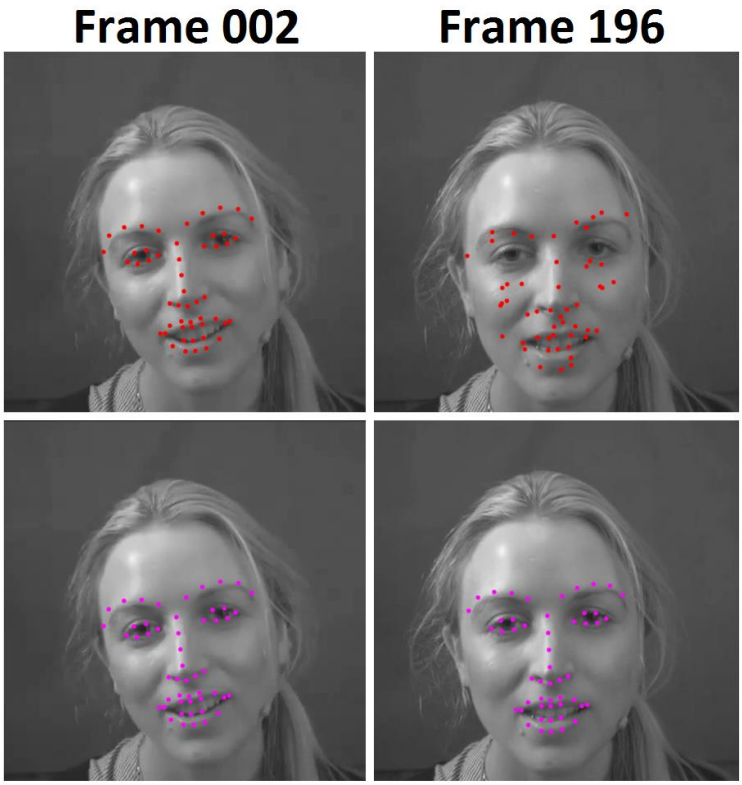

Figura 4.2 Comportamiento del tracker según la estrategia de inicialización de los landmarks.

En la Figura 4.3 se muestra el histograma de error acumulado calculado sobre un total de 2.512 frames y normalizado respecto a la distancia interocular en cada frame. Se observa como el error normalizado al inicializarlo con los landmarks calculados en el frame anterior, es mucho mayor que al inicializarlo utilizando el bounding box; en el segundo caso el 20% de las imágenes tienen un error mayor de 0.002 mientras que en el primero, el 20% de las imágenes tienen un error mayor de 0.025.

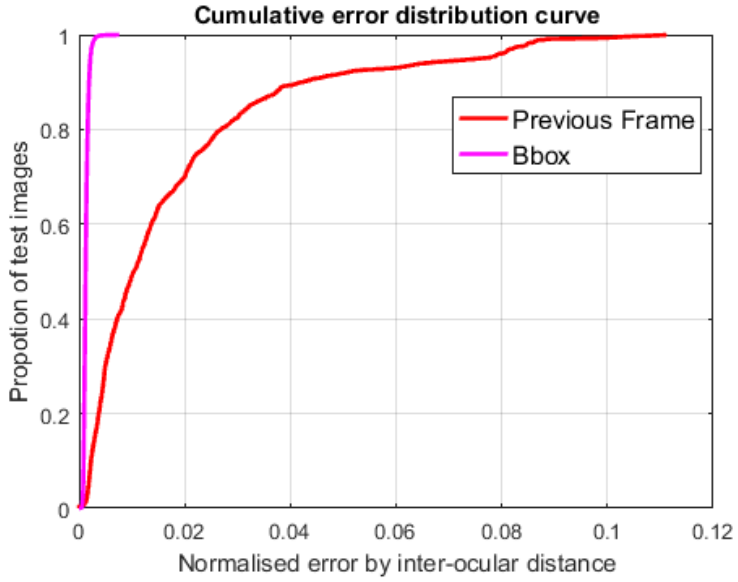

Figura 4.3 Histograma de error acumulado en el caso de inicializar a partir del bounding box (magenta) o de los landmarks obtenidos en el frame anterior (rojo).

4.1.1.3 Regresores generados

Una vez visto que, pese a lo que pueda parecer a priori, la mejor estrategia de inicialización es utilizar los landmarks medios, se decidió utilizar ésta a la hora de entrenar nuevos regresores. El entrenamiento de los nuevos regresores se realizó utilizando el SSDM C++ con el fin de emplearlos en el software programado en ese lenguaje.

Cuando se decidió realizar el entrenamiento de nuevos regresores, únicamente se disponía de 16 GB de RAM por lo que los primeros regresores no se pudieron entrenar con la cantidad de imágenes deseada. Más tarde se dispuso de 64 GB de RAM y se pudo entrenar con un número mayor de imágenes.

Los regresores que se han generado son los siguientes:

- **SSDR_1157:** Primer regresor entrenado. Se utilizaron únicamente la mitad de las imágenes de las bases de datos AFW y HELEN. La razón de entrenar este número de imágenes es por la limitación de RAM ya comentada.

- **SSDR:** Como ya se ha comentado, es el regresor entrenado por la Universidad de Surrey e incluido en el software SSDM C++.

- **MIX_1157_R_80:** Regresor entrenado con las mismas imágenes reales que el SSDR_1157. Además, se utilizaron 80 frames seleccionados de los vídeos 04 de cuatro usuarios de la base de datos sintética (rotación roll, 20 frames por usuario).

- **MIX_3283_R_140:** Regresor entrenado con las mismas imágenes reales que el SSDR y 140 frames seleccionados de los vídeos 04 de cuatro usuarios de la base de datos sintética (rotación roll, 35 frames por usuario).

- **MIX_1157_YP_1200:** Regresor entrenado con las mismas imágenes reales que el SSDR_1157 y 1200 frames seleccionados de los vídeos 05 y 06 de todos los usuarios de la base de datos sintética (rotaciones yaw y pitch, 60 frames por usuario y vídeo).

- **MIX_3283_YP_1200:** Regresor entrenado con las mismas imágenes reales que el SSDR y 1200 frames seleccionados de los vídeos 05 y 06 de todos los usuarios de la base de datos sintética (rotaciones yaw y pitch, 60 frames por usuario y vídeo).

En cuanto a la anchura W de los regresores, todos tienen la misma que el original (6 hilos). En la Tabla 4.1 se presenta un resumen de los regresores entrenados en el que se detalla el número de imágenes reales y sintéticas que se han utilizado para cada regresor, así como las bases de datos reales empleadas y las rotaciones presentes en los frames sintéticos seleccionados.

	IMÁGENES REALES		IMÁGENES SINTÉTICAS			
NOMBRE	**Nº**	**BASES DE DATOS**	**Nº**	**ROLL**	**YAW**	**PITCH**
SSDR_1157	**1157**	AFW, HELEN	✘	✘	✘	✘
SSDR	**3283**	AFW, HELEN, IBUG, LFPW	✘	✘	✘	✘
MIX_1157_R_80	**1157**	AFW, HELEN	**80**	✓	✘	✘
MIX_3283_R_140	**3283**	AFW, HELEN, IBUG, LFPW	**140**	✓	✘	✘
MIX_1157_YP_1200	**1157**	AFW, HELEN	**1200**	✘	✓	✓
MIX_3283_YP_1200	**3283**	AFW, HELEN, IBUG, LFPW	**1200**	✘	✓	✓

Tabla 4.1 Resumen de los regresores entrenados.

4.1.2 Rotación en roll de los frames

La idea de esta técnica es realizar una rotación de la imagen de manera que la cabeza siempre se encuentre con valores bajos de rotación roll. De esta forma, el sistema de seguimiento no presenta problemas cuando la imagen original tiene un alto valor de rotación roll.

El algoritmo implementado para realizar la rotación de los frames se detalla a continuación:

Algoritmo 2 Rotación en roll de los frames

1: Inicializamos con valor $Roll = 0$
2: Mientras tenga frames **repito:**
3: Tomo el siguiente frame
4: Detecto el bounding box
5: Roto la imagen $-Roll$ grados
6: Roto el bounding box $-Roll$ grados
7: Detecto los landmarks sobre la imagen rotada
8: Roto los landmarks $Roll$ grados
9: Utilizando el método propio de SSDM, calculo la rotación roll y actualizo $Roll$
10: Guardo los landmarks
11: Fin de la repetición
12: Fin del algoritmo

En la parte de arriba de la Figura 4.4 se puede observar un frame en el que se presenta un alto valor de rotación. En la parte de abajo, se observa el mismo frame rotado mediante el algoritmo 2.

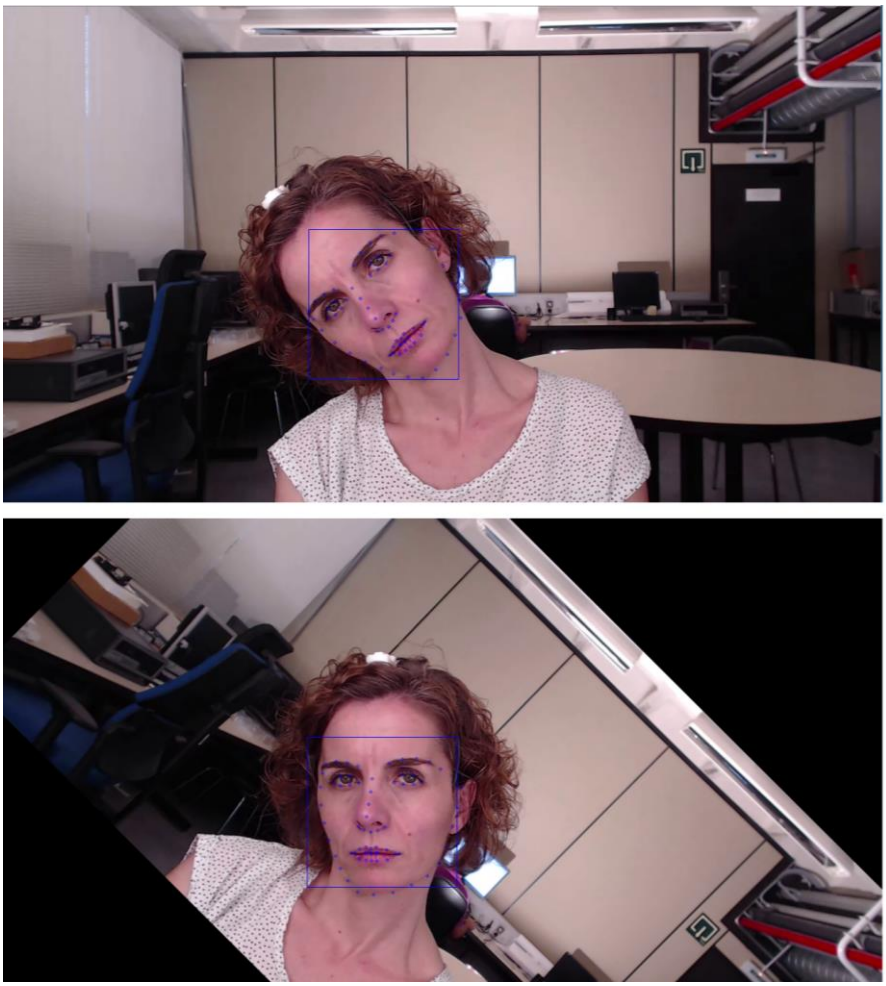

Figura 4.4 Frame original (arriba) y frame rotado (abajo).

Se puede ver como, por mucho valor de rotación roll que presente el frame original, al hacer la detección de los landmarks en la imagen rotada, el sistema de seguimiento funcionará correctamente.

Al utilizar este algoritmo, no hace falta que el regresor esté entrenado para altos valores de rotación roll, es por eso que los regresores *MIX_1157_YP_1200* y *MIX_1157_YP_1200* no han sido entrenados con frames sintéticos que presenten un alto valor de rotación roll.

4.2 Ciclos en la detección de los landmarks

Como ya se ha comentado en el apartado 3.2.1, una característica de los sistemas de seguimiento es la presencia de *jitter*. Se ha realizado una prueba para medir el jitter existente en IF y en SSDM. La prueba consiste en generar un vídeo de 100 fotogramas iguales y analizar los landmarks obtenidos en cada fotograma. Mirando los ficheros en los que se guardan los landmarks detectados en cada frame, se ha visto que, en el caso del SSDM, en algunas ocasiones no existe jitter (los landmarks son iguales para todos los frames) y en otras la localización de los landmarks varía cíclicamente entre 2-3 valores.

Esta característica del SSDM no es nada interesante ya que, a la hora de calcular los landmarks 3D por retroproyección, interesa obtener la posición media para minimizar el error entre los landmarks 3D calculados y la detección de los landmarks en cada frame.

Para solucionar este problema, se realizó una modificación en el software de SSDM que consiste en, a la hora de calcular el bounding box, aplicarle un factor aleatorio que modifique la posición de su centro y su tamaño. Esta modificación hace que la inicialización de los landmarks sea distinta en cada iteración, consiguiendo que los landmarks detectados varíen ligeramente y permitiendo realizar correctamente la media de la posición detectada en 100 frames.

Esta modificación también se utiliza a la hora de calcular la pose en el primer frame y poder calcular el valor de pose zeroed con mayor precisión.

Capítulo 5
Resultados

En este capítulo se presentan los resultados del error de estimación de pose obtenidos utilizando los distintos sistemas de tracking y modelos de cabeza 3D, y aplicando las mejoras comentadas en los capítulos anteriores.

En primer lugar se indica la nomenclatura utilizada en la presentación de resultados así como ciertas decisiones iniciales tomadas a fin de facilitar tanto la obtención como la comprensión de los mismos. A continuación se aborda la presentación de resultados, comenzando por los resultados obtenidos utilizando el software en bruto (sin ninguna mejora ni implementación) y prosiguiendo con los resultados obtenidos al aplicar las mejoras propuestas en el trabajo.

El apartado final está dedicado a presentar, a modo de resumen, la comparativa entre los errores de HPE obtenidos por los métodos originales, los métodos propios del software utilizado y el método con mejores resultados obtenidos en este proyecto.

5.1 Nomenclatura y decisiones iniciales

La nomenclatura utilizada en la presentación de resultados es la siguiente:

- Siempre que se utilice el término *SFM* o *BFM* se referirá al modelo medio; si se hace referencia a los modelos particulares generados con el software 4DFace, se utilizara el término *Reconstructed*; si por el contrario se hace referencia a los modelos sintéticos originales con los que se ha realizado la base de datos *UPNA Synthetic Head Pose Database*, se utilizará el termino *Groundtruth*.

- Como ya se ha comentado, al realizar la retroproyección se puede obtener tanto landmarks 3D como vértices 3D. El término utilizado cuando se usen los primeros será *Landmarks 3D* mientras que con los segundos, se utilizará el término *Vertex 3D.*

- Al regresor de IntraFace se le denomina *IF_126*; se ha escogido este nombre ya que se trata del regresor incluido en la versión 1.26 del software. El nombre de los regresores utilizados en el software SSDM es el especificado en la Tabla 4.1.

En cuanto a las decisiones iniciales:

- A no ser que se especifique lo contrario, los resultados siempre se obtienen utilizando la base de datos *UPNA Head Pose Database.*

- A no ser que se especifique lo contrario, los resultados se obtienen utilizando, para el caso de modelos *SFM*, *BFM* y *Groundtruth*, los índices *Landmarks 3D*; y para el caso de modelos *Reconstructed*, los índices *Vertex 3D* del modelo *SFM*.

- Los resultados representan la media y la desviación típica ($\mu \pm \sigma$) del error absoluto entre pose estimada y pose groundtruth calculado para todos los vídeos de todos los usuarios. Los errores se miden: en traslación, en milímetros; y en rotación, en grados.

5.2 Regresores originales

La primera prueba realizada es la comparación entre utilizar como sistemas de seguimiento el software de IntraFace o el SSDM con los regresores incluidos y como modelos geométricos 3D el SFM o el BFM sin aplicar ninguna de las implementaciones comentadas a lo largo del trabajo.

En la Tabla 5.1 se muestran los resultados obtenidos para cada combinación de sistema de seguimiento y modelo geométrico 3D.

	IF_126		**SSDR**	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	12.913 ± 18.38	12.541 ± 18.34	13.098 ± 18.19	12.720 ± 18.29
Ty(mm)	166.626 ± 10.71	168.247 ± 10.61	166.092 ± 10.95	167.993 ± 10.91
Tz(mm)	47.510 ± 29.02	39.163 ± 25.25	54.539 ± 29.78	43.925 ± 24.55
T Mean(mm)	75.683 ± 69.04	73.317 ± 70.61	77.909 ± 67.97	74.879 ± 69.64
Roll(º)	0.747 ± 0.52	0.695 ± 0.54	0.905 ± 0.92	0.858 ± 0.92
Yaw(º)	1.648 ± 1.30	1.839 ± 1.41	1.342 ± 1.18	1.879 ± 1.30
Pitch(º)	4.189 ± 3.24	3.802 ± 3.33	4.194 ± 3.63	3.682 ± 3.24
R Mean(º)	2.195 ± 2.51	2.112 ± 2.47	2.147 ± 2.69	2.140 ± 2.39

Tabla 5.1 Resultados de error de HPE utilizando los regresores IF_126 y SSDR y los modelos geométricos BFM y SFM.

Como ya se ha dicho en el apartado 3.1.3, los sistemas de coordenadas de la base de datos y de los modelos 3D son diferentes. Esto genera errores sistemáticos tanto en traslación como en rotación.

5.2.1 Zeroed

La primera implementación es referenciar la pose respecto al primer frame (hacer un *zeroed*) tanto en los valores groundtruth como en la estimación para eliminar los errores sistemáticos debidos a la diferencia entre los sistemas de coordenadas.

En la Tabla 5.2 se muestran los resultados obtenidos al realizar esta implementación. Se observa como el error disminuye notablemente tanto en traslación como en rotación.

	IF_126		**SSDR**	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	10.026 ± 9.41	10.550 ± 9.86	10.371 ± 12.42	10.832 ± 11.40
Ty(mm)	12.805 ± 12.71	14.128 ± 13.36	13.608 ± 13.93	15.164 ± 15.45
Tz(mm)	5.825 ± 6.22	7.039 ± 7.61	5.816 ± 8.57	7.166 ± 10.20
T Mean(mm)	9.552 ± 10.22	10.572 ± 10.94	9.932 ± 12.28	11.054 ± 12.97
Roll(º)	0.419 ± 0.42	0.426 ± 0.46	0.615 ± 0.92	0.590 ± 0.88
Yaw(º)	0.897 ± 0.82	0.930 ± 0.82	0.932 ± 1.08	0.976 ± 1.07
Pitch(º)	1.291 ± 1.29	1.367 ± 1.34	1.390 ± 1.46	1.484 ± 1.59
R Mean(º)	0.869 ± 0.98	0.908 ± 1.02	0.979 ± 1.22	1.017 ± 1.27

Tabla 5.2 Resultados de error de HPE utilizando los regresores IF_126 y SSDR y los modelos geométricos BFM y SFM y aplicando zeroed.

5.2.1.1 Zeroed average

Una variación del zeroed es el zeroed average; que consiste en calcular la pose del primer frame a partir de la media de los landmarks obtenidos al repetir el primer frame 100 veces. Esta variación se realiza para evitar errores en caso de no calcular la pose del primer frame correctamente.

	IF_126		**SSDR**	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	9.659 ± 9.59	10.198 ± 9.96	10.755 ± 12.41	11.179 ± 11.36
Ty(mm)	12.083 ± 12.16	13.171 ± 12.61	14.016 ± 14.51	15.795 ± 16.05
Tz(mm)	5.665 ± 6.19	6.858 ± 7.59	5.868 ± 8.66	7.221 ± 10.34
T Mean(mm)	9.136 ± 9.99	10.076 ± 10.58	10.213 ± 12.56	11.399 ± 13.30
Roll(º)	0.383 ± 0.40	0.382 ± 0.44	0.613 ± 0.93	0.587 ± 0.88
Yaw(º)	0.852 ± 0.84	0.887 ± 0.83	0.977 ± 1.08	1.022 ± 1.06
Pitch(º)	1.216 ± 1.24	1.273 ± 1.27	1.425 ± 1.52	1.527 ± 1.65
R Mean(º)	0.817 ± 0.96	0.847 ± 0.98	1.005 ± 1.25	1.046 ± 1.30

Tabla 5.3 Resultados de error de HPE utilizando los regresores IF_126 y SSDR y los modelos geométricos BFM y SFM y aplicando zeroed average.

Comparando los resultados obtenidos por esta implementación (Tabla 5.3) con los obtenidos sin aplicar el promediado (Tabla 5.2) se observa que, en el caso de IntraFace los resultados mejoran y en el caso de SSDM no. Esto puede deberse a que el software de IntraFace realiza la detección de los landmarks peor en el primer frame y, realizar un promediado le ayuda a estimar mejor la pose inicial.

Aunque en el caso del software SSDM no se consiga una mejora, se ha decidido utilizar el zeroed average por la seguridad que aporta ante un fallo en la estimación de la pose del primer frame.

5.2.2 Uso de Vertex 3D

Como ya se ha dicho en el apartado 3.2.2, a la hora de calcular la correspondencia entre los landmarks 2D y los puntos del modelo 3D mediante retroproyección, se puede obtener tanto los landmarks 3D como los vértices 3D. En la Tabla 5.4 se muestran los resultados utilizando como puntos del modelo 3D los vértices 3D.

	IF_126		**SSDR**	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	9.588 ± 9.44	10.059 ± 9.75	10.689 ± 12.30	11.271 ± 11.41
Ty(mm)	12.069 ± 12.12	13.093 ± 12.54	14.034 ± 14.53	15.937 ± 16.24
Tz(mm)	5.623 ± 6.14	6.845 ± 7.53	5.766 ± 8.58	7.178 ± 10.35
T Mean(mm)	9.093 ± 9.91	9.999 ± 10.46	10.163 ± 12.52	11.462 ± 13.41
Roll(º)	0.389 ± 0.41	0.383 ± 0.45	0.603 ± 0.92	0.574 ± 0.87
Yaw(º)	0.849 ± 0.83	0.876 ± 0.81	0.974 ± 1.07	1.028 ± 1.07
Pitch(º)	1.216 ± 1.24	1.269 ± 1.27	1.426 ± 1.52	1.535 ± 1.66
R Mean(º)	0.818 ± 0.95	0.843 ± 0.98	1.001 ± 1.24	1.046 ± 1.31

Tabla 5.4 Resultados de error de HPE utilizando vertex 3D, los regresores IF_126 y SSDR y los modelos geométricos BFM y SFM.

Comparando los resultados de la Tabla 5.4 con los de la Tabla 5.3, se observa que no existen grandes diferencias entre trabajar con landmarks 3D o con vertex 3D. Se ha decidido utilizar los landmarks 3D por ser más ajustados a la realidad.

5.2.3 Modelos reconstruidos

En el apartado 3.3 se ha presentado una herramienta dentro del software de SSDM que permite generar modelos 3D adaptados a cada usuario. En la Tabla 5.5 se realiza una comparación entre los resultados obtenidos utilizando el modelo medio SFM y los modelos reconstruidos para cada usuario (obtenidas deformando el modelo SFM). Como ya se ha dicho, al utilizar los modelos reconstruidos se utilizan los vértices 3D calculados para el modelo SFM, pero en el apartado anterior se ha visto que no hay diferencias significativas en el HPE utilizando landmarks 3D o vértices 3D; por lo que se puede realizar una comparación entre los dos modelos.

	IF_126		SSDR	
	SFM	**Reconstructed**	**SFM**	**Reconstructed**
Tx(mm)	10.198 ± 9.96	9.637 ± 9.95	11.179 ± 11.36	10.424 ± 11.28
Ty(mm)	13.171 ± 12.61	11.946 ± 11.33	15.795 ± 16.05	14.879 ± 15.35
Tz(mm)	6.858 ± 7.59	6.706 ± 6.95	7.221 ± 10.34	7.097 ± 10.15
T Mean(mm)	10.076 ± 10.58	9.430 ± 9.82	11.399 ± 13.30	10.800 ± 12.86
Roll(º)	0.382 ± 0.44	0.396 ± 0.43	0.587 ± 0.88	0.639 ± 0.94
Yaw(º)	0.887 ± 0.83	1.067 ± 1.26	1.022 ± 1.06	1.068 ± 1.27
Pitch(º)	1.273 ± 1.27	1.215 ± 1.24	1.527 ± 1.65	1.496 ± 1.59
R Mean(º)	0.847 ± 0.98	0.893 ± 1.11	1.046 ± 1.30	1.068 ± 1.34

Tabla 5.5 Resultados de error de HPE utilizando los regresores IF_126 y SSDR y los modelos geométricos SFM y Reconstructed.

Fijándose en la Tabla 5.5, se observa que, a pesar de lo que pudiera parecer en un principio, utilizar los modelos reconstruidos no mejora la estimación respecto a utilizar el modelo medio SFM, es más, la empeora. Por lo tanto, no haremos uso de los modelos reconstruidos.

5.2.4 Base de datos sintética

Si en vez de utilizar la base de datos real, se utiliza la sintética, se obtienen los resultados mostrados en la Tabla 5.6.

	IF 126		SSDR	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	11.383 ± 12.23	10.393 ± 10.31	10.566 ± 20.31	11.136 ± 21.76
Ty(mm)	10.827 ± 11.13	9.189 ± 8.73	13.705 ± 23.21	13.645 ± 25.99
Tz(mm)	4.180 ± 4.34	4.778 ± 5.14	15.932 ± 105.06	17.522 ± 109.62
T Mean(mm)	8.797 ± 10.40	8.120 ± 8.69	13.401 ± 63.25	14.101 ± 66.30
Roll(º)	0.301 ± 0.32	0.303 ± 0.30	0.735 ± 2.99	0.701 ± 2.98
Yaw(º)	1.031 ± 1.15	0.958 ± 1.03	1.039 ± 2.05	1.064 ± 2.11
Pitch(º)	1.058 ± 1.20	0.987 ± 1.01	1.445 ± 2.62	1.406 ± 2.89
R Mean(º)	0.797 ± 1.04	0.750 ± 0.91	1.073 ± 2.60	1.057 ± 2.70

Tabla 5.6 Resultados de error de HPE sobre la base de datos sintética utilizando los regresores IF_126 y SSDR y los modelos geométricos BFM y SFM.

Comparando la Tabla 5.3 y la Tabla 5.6, se puede observar:

1. En el caso de utilizar IntraFace como sistema de seguimiento, el error medio disminuye. Llama la atención que, sabiendo que las cabezas generadas para realizar la base de datos sintética son deformaciones del modelo BFM, utilizar el modelo SFM proporcione una ligera mejoría respecto a utilizar el BFM.

2. En el caso de utilizar SSDM con el regresor SSDR, el error medio es similar al obtenido utilizando la base de datos real. Sin embargo, los valores de desviación aumentan en gran medida.

En ambos casos utilizar el modelo SFM o el BFM proporciona resultados similares, por lo tanto, parece ser que el gran aumento de los valores de desviación es debido al sistema de seguimiento utilizado. Para comprobarlo se ha decidido realizar un experimento en el que se utilicen los modelos groundtruth con los que se ha creado la base de datos sintética.

5.2.4.1 Modelos sintéticos groundtruth

La prueba realizada se basa en calcular el error de HPE sobre la base de datos sintética, utilizando como modelos 3D los modelos groundtruth utilizados para generar esta base de datos. De esta forma, al no existir errores en el modelo 3D, se puede valorar el error de estimación propio de los sistemas de seguimiento.

En la Tabla 5.7 se muestran los resultados obtenidos al realizar este experimento.

	IF 126	**SSDR**
	Groundtruth	**Groundtruth**
Tx(mm)	10.003 ± 9.81	8.332 ± 19.42
Ty(mm)	9.171 ± 9.15	11.785 ± 23.00
Tz(mm)	3.085 ± 3.11	14.225 ± 101.76
T Mean(mm)	7.420 ± 8.53	11.447 ± 61.32
Roll(º)	0.267 ± 0.25	0.706 ± 3.01
Yaw(º)	0.954 ± 0.96	0.805 ± 2.02
Pitch(º)	0.881 ± 0.94	1.257 ± 2.72
R Mean(º)	0.701 ± 0.85	0.923 ± 2.63

Tabla 5.7 Comparativa del error de HPE entre los regresores IF_126 y SSDR utilizando la base de datos sintética y el modelo geométrico groundtruth.

Se puede observar que los resultados son similares a los obtenidos en la Tabla 5.6 por lo que se concluye que el problema reside en el sistema de seguimiento.

Para tener una mayor visión de lo que está sucediendo, en la Figura 5.1 se representan los resultados de forma gráfica haciendo uso de diagramas de cajas. Los resultados se han agrupado según usuarios (izquierda) y según vídeos (derecha). Se puede ver la existencia de una gran cantidad de valores atípicos presente en los vídeos 04 (vídeo de rotación roll) de los usuarios 04, 05 y 06, llegando a errores de 140 cm en el caso de traslación y de 50º en el de rotación. Llama la atención que los valores atípicos se concentren en los vídeos 04 por lo que se ha decidido representar la respuesta del regresor SSDR ante estos vídeos.

En la Figura 5.2 se representa la respuesta de los regresores SSDR (izquierda, landmarks verdes) e IF_126 (derecha, landmarks amarillos) ante el frame 190 del vídeo 04 de los usuarios sintéticos 04 (arriba) y 10 (abajo). Se puede ver como el regresor SSDR se "pierde" completamente en estos vídeos mientras que el IF_126 realiza una buena detección.

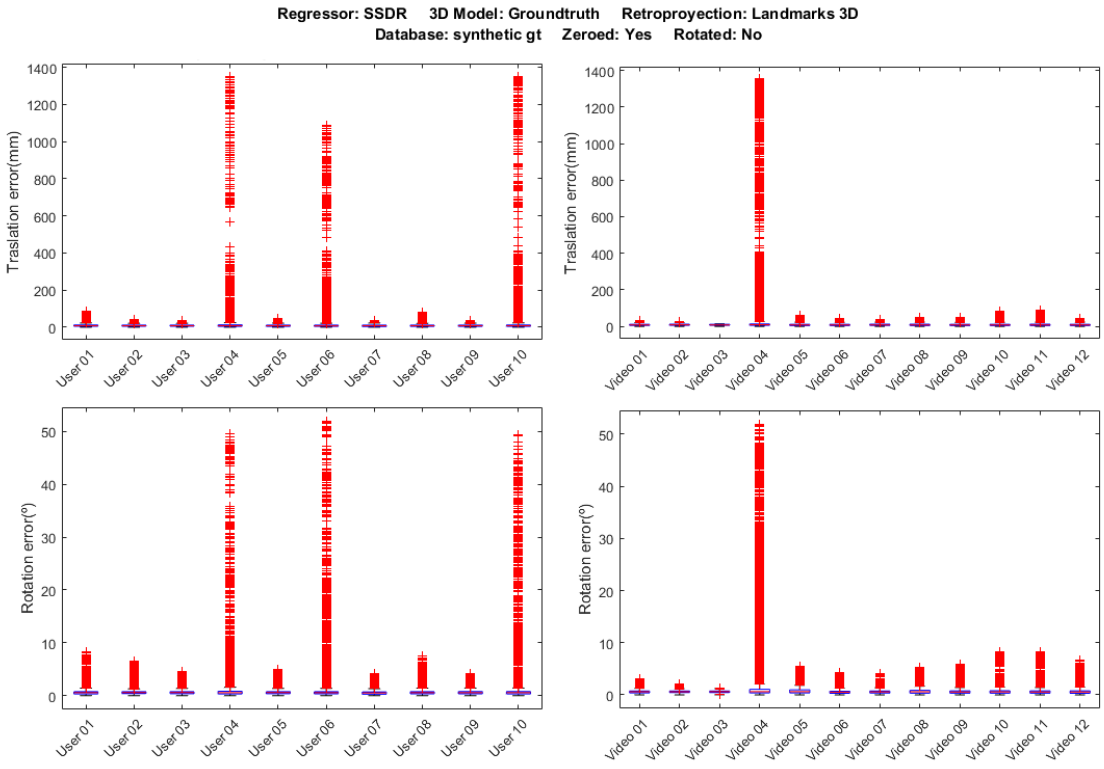

Figura 5.1 Diagrama de cajas del error obtenido sobre la base de datos sintética utilizando el sistema de seguimiento SSDR y el modelo geométrico groundtruth. El error se agrupa por usuarios (izquierda) o por vídeos (derecha).

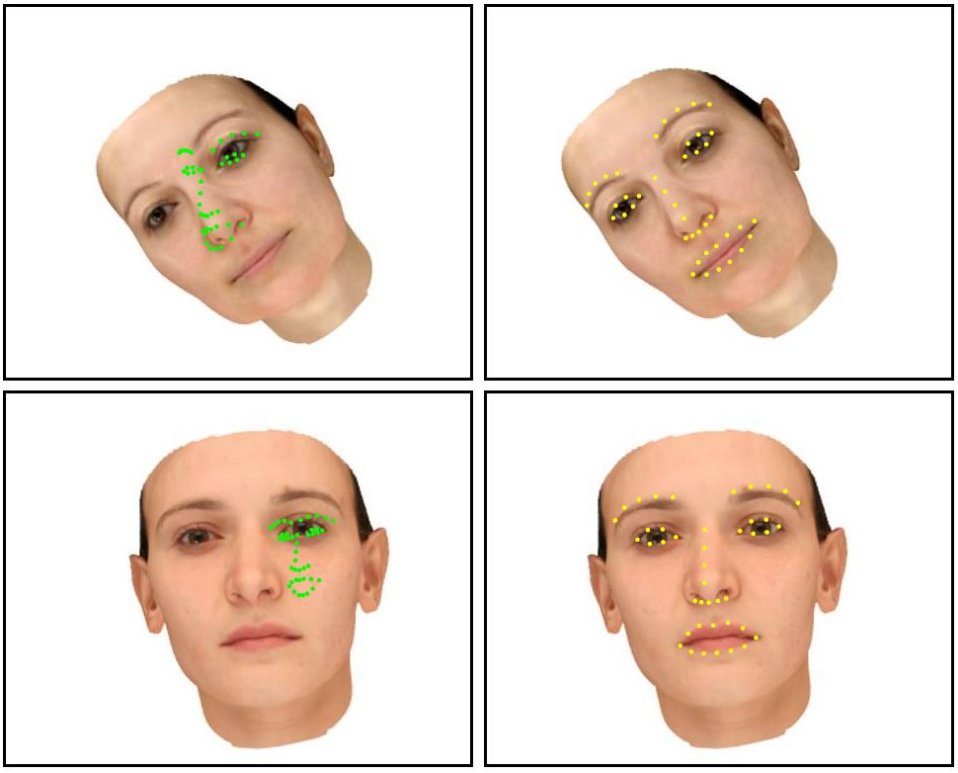

Figura 5.2 Respuesta de los regresores SSDR (izquierda) e IF_126 (derecha) en vídeos 04 de la base de datos sintética.

En la Tabla 5.8 se muestran los resultados obtenidos al realizar el mismo experimento eliminando los vídeos problemáticos (vídeos 04 de los usuarios 04, 05 y 06).

	IF 126	SSDR
	Groundtruth	Groundtruth
Tx(mm)	9.678 ± 9.82	6.471 ± 7.59
Ty(mm)	8.889 ± 9.22	9.386 ± 9.30
Tz(mm)	2.994 ± 3.13	3.361 ± 3.64
T Mean(mm)	7.187 ± 8.52	6.406 ± 7.65
Roll(º)	0.254 ± 0.25	0.442 ± 0.55
Yaw(º)	0.923 ± 0.96	0.609 ± 0.76
Pitch(º)	0.853 ± 0.94	0.968 ± 1.01
R Mean(º)	0.677 ± 0.85	0.673 ± 0.82

Tabla 5.8 Comparativa del error de HPE entre los regresores IF_126 y SSDR utilizando la base de datos sintética sin los vídeos problemáticos y el modelo geométrico groundtruth.

Comparando la Tabla 5.7 y la Tabla 5.8 se puede ver que, en el caso del regresor IF_126, los resultados no presentan grandes cambios mientras que, en el caso del regresor SSDR, tanto la media como la desviación disminuyen considerablemente (hasta el punto de igualar los resultados de IF_126).

En la Figura 5.3 se muestran los diagramas de cajas de error de estimación HPE del regresor SSDR, obtenidos al eliminar los vídeos problemáticos. Se observa que, a pesar de la presencia de valores atípicos, los errores no alcanzan los valores representados en la Figura 5.2.

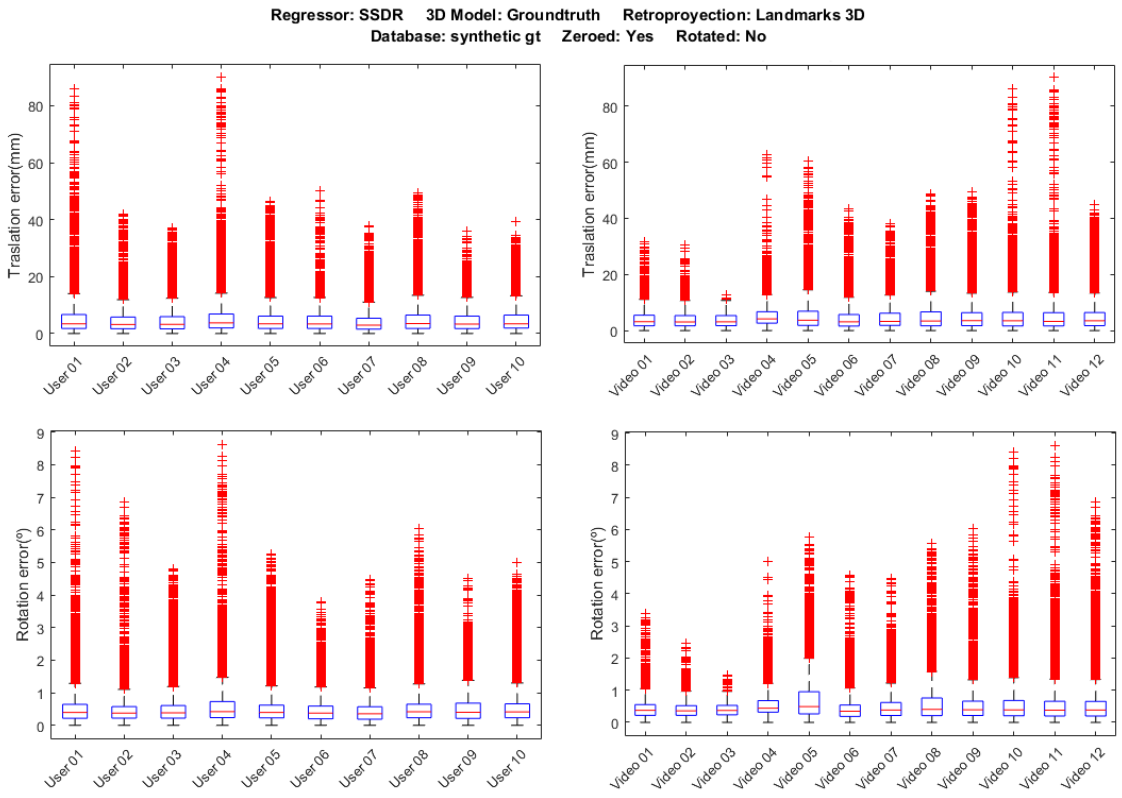

Figura 5.3 Diagrama de cajas del error obtenido sobre la base de datos sintética sin los vídeos problemáticos utilizando el sistema de seguimiento SSDR y el modelo geométrico groundtruth y. El error se agrupa por usuarios (izquierda) o por vídeos (derecha).

5.3 Rotación de los frames

Como ya se ha comentado en el apartado 4.1, una de las soluciones implementadas para resolver el problema de la pérdida del sistema de seguimiento, es realizar una rotación de la imagen a fin de que el regresor trabaje sobre imágenes con valores bajos de rotación roll. Se ha realizado una comparación entre los resultados obtenidos al utilizar o no el algoritmo de rotación sobre la base de datos real (Tabla 5.9) y sobre la base de datos sintética (Tabla 5.10).

5.3.1 Base de datos real

	SSDR NO ROTATED		SSDR ROTATED	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	10.755 ± 12.41	11.179 ± 11.36	10.322 ± 11.44	9.729 ± 9.14
Ty(mm)	14.016 ± 14.51	15.795 ± 16.05	13.978 ± 13.64	15.467 ± 13.86
Tz(mm)	5.868 ± 8.66	7.221 ± 10.34	5.340 ± 5.28	6.615 ± 7.23
T Mean(mm)	10.213 ± 12.56	11.399 ± 13.30	9.880 ± 11.29	10.604 ± 11.08
Roll(º)	0.613 ± 0.93	0.587 ± 0.88	0.622 ± 0.77	0.572 ± 0.66
Yaw(º)	0.977 ± 1.08	1.022 ± 1.06	0.885 ± 0.97	0.876 ± 0.83
Pitch(º)	1.425 ± 1.52	1.527 ± 1.65	1.423 ± 1.44	1.487 ± 1.49
R Mean(º)	1.005 ± 1.25	1.046 ± 1.30	0.977 ± 1.15	0.978 ± 1.12

Tabla 5.9 Comparativa del error de HPE entre usar o no el algoritmo de rotación con el regresor SSDR y los modelos geométricos BFM y SFM.

Los resultados obtenidos implementando el algoritmo de rotación no distan mucho de los obtenidos sin dicha implementación. Esto es debido a que, en el caso de la base de datos real, el regresor no se pierde de la manera en que lo hace con la base de datos sintética.

5.3.2 Base de datos sintética

	SSDR NO ROTATED		SSDR ROTATED	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	10.566 ± 20.31	11.136 ± 21.76	8.997 ± 11.39	8.087 ± 8.71
Ty(mm)	13.705 ± 23.21	13.645 ± 25.99	12.412 ± 11.80	11.780 ± 11.28
Tz(mm)	15.932 ± 105.06	17.522 ± 109.62	5.004 ± 5.18	6.091 ± 6.58
T Mean(mm)	13.401 ± 63.25	14.101 ± 66.30	8.804 ± 10.38	8.653 ± 9.37
Roll(º)	0.735 ± 2.99	0.701 ± 2.98	0.527 ± 0.60	0.461 ± 0.51
Yaw(º)	1.039 ± 2.05	1.064 ± 2.11	0.827 ± 1.12	0.781 ± 0.91
Pitch(º)	1.445 ± 2.62	1.406 ± 2.89	1.298 ± 1.26	1.182 ± 1.12
R Mean(º)	1.073 ± 2.60	1.057 ± 2.70	0.884 ± 1.08	0.808 ± 0.93

Tabla 5.10 Comparativa del error de HPE entre usar o no el algoritmo de rotación sobre la base de datos sintética con el regresor SSDR y los modelos geométricos BFM y SFM.

En este caso, la mejora resulta evidente tanto en términos de media como de desviación. Comparando estos resultados con los de la Tabla 5.6, se observa que los valores obtenidos por el regresor SSDR al realizar la rotación se asemejan a los obtenidos por el regresor IF_126.

5.3.2.1 Modelos sintéticos groundtruth

Si se calcula el error de HPE sobre la base de datos sintética utilizando como modelos 3D los modelos groundtruth, obtenemos los resultados presentados en la Tabla 5.11.

	SSDR NO ROTATED	SSDR ROTATED
	Groundtruth	Groundtruth
Tx(mm)	8.332 ± 19.42	6.682 ± 7.41
Ty(mm)	11.785 ± 23.00	10.290 ± 9.27
Tz(mm)	14.225 ± 101.76	3.604 ± 3.64
T Mean(mm)	11.447 ± 61.32	6.859 ± 7.67
Roll(º)	0.706 ± 3.01	0.483 ± 0.55
Yaw(º)	0.805 ± 2.02	0.624 ± 0.73
Pitch(º)	1.257 ± 2.72	1.064 ± 1.02
R Mean(º)	0.923 ± 2.63	0.724 ± 0.83

Tabla 5.11 Comparativa del error de HPE entre usar o no el algoritmo de rotación sobre la base de datos sintética con el regresor SSDR y el modelo geométrico Groundtruth.

Al igual que en el caso anterior, la mejora resulta evidente tanto en términos de media como de desviación. Además, mirando la Figura 5.4 se ve que, a pesar de la presencia de valores atípicos, los errores no alcanzan los valores obtenidos sin implementar la rotación.

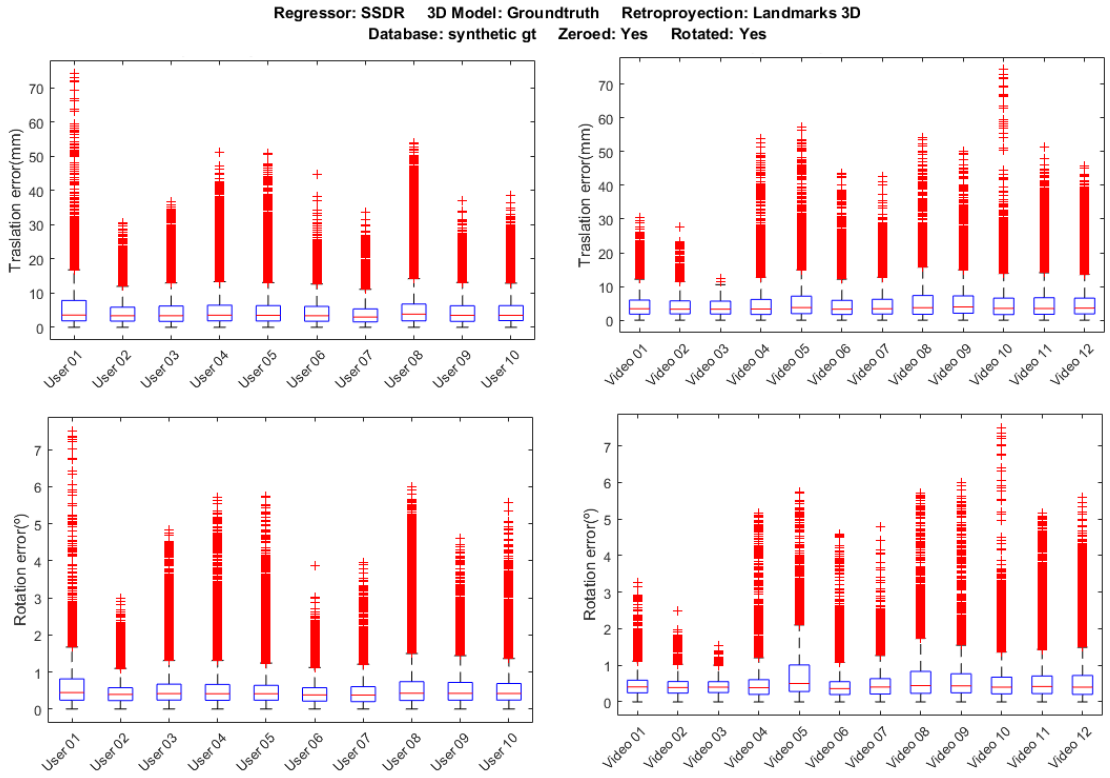

Figura 5.4 Diagrama de cajas del error obtenido sobre la base de datos sintética aplicando el algoritmo de rotación y utilizando el sistema de seguimiento SSDR y el modelo geométrico groundtruth. El error se agrupa por usuarios (izquierda) o por vídeos (derecha).

5.4 Regresores entrenados

La otra solución implementada para resolver el problema de la pérdida del sistema de seguimiento, es entrenar regresores más robustos frente a altos valores de rotación. En la Tabla 5.12, Tabla 5.13 y Tabla 5.14 se muestran los resultados obtenidos por el conjunto de regresores entrenados. En este caso todos los resultados se calculan utilizando la base de datos real ya que, al haber utilizado la base de datos sintética para realizar el entrenamiento, no se pueden obtener resultados sobre la misma.

	SSDR_1157		**SSDR**	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	10.587 ± 15.23	11.934 ± 16.04	10.755 ± 12.41	11.179 ± 11.36
Ty(mm)	14.825 ± 17.87	18.918 ± 21.36	14.016 ± 14.51	15.795 ± 16.05
Tz(mm)	9.195 ± 53.09	11.029 ± 55.66	5.868 ± 8.66	7.221 ± 10.34
T Mean(mm)	11.535 ± 33.60	13.960 ± 35.82	10.213 ± 12.56	11.399 ± 13.30
Roll(º)	0.620 ± 1.78	0.663 ± 1.78	0.613 ± 0.93	0.587 ± 0.88
Yaw(º)	0.906 ± 1.36	1.057 ± 1.51	0.977 ± 1.08	1.022 ± 1.06
Pitch(º)	1.519 ± 1.85	1.822 ± 2.14	1.425 ± 1.52	1.527 ± 1.65
R Mean(º)	1.015 ± 1.72	1.180 ± 1.89	1.005 ± 1.25	1.046 ± 1.30

Tabla 5.12 Comparativa del error de HPE entre usar el regresor SSDR_1157 o el regresor SSDR.

	MIX_1157_R_80		**MIX_3283_R_140**	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	10.149 ± 13.23	12.436 ± 13.74	10.040 ± 11.67	11.012 ± 11.45
Ty(mm)	15.039 ± 15.13	19.212 ± 18.39	13.689 ± 13.61	15.985 ± 15.32
Tz(mm)	6.073 ± 7.43	7.651 ± 9.74	5.253 ± 7.55	6.803 ± 9.26
T Mean(mm)	10.421 ± 12.90	13.100 ± 15.16	9.661 ± 11.75	11.266 ± 12.83
Roll(º)	0.483 ± 0.70	0.517 ± 0.72	0.610 ± 0.80	0.591 ± 0.73
Yaw(º)	0.837 ± 1.01	1.028 ± 1.18	0.902 ± 1.00	1.001 ± 1.07
Pitch(º)	1.560 ± 1.63	1.826 ± 1.86	1.392 ± 1.44	1.530 ± 1.56
R Mean(º)	0.960 ± 1.26	1.124 ± 1.44	0.968 ± 1.16	1.040 ± 1.23

Tabla 5.13 Comparativa del error de HPE entre usar el regresor MIX_1157_R_80 o el regresor MIX_3283_R_140.

	MIX_1157_YP_1200		**MIX_3283_YP_1200**	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	9.720 ± 15.52	11.898 ± 17.28	9.384 ± 10.79	10.530 ± 11.21
Ty(mm)	13.220 ± 13.39	16.996 ± 16.89	13.697 ± 14.30	15.660 ± 16.09
Tz(mm)	7.014 ± 25.99	8.633 ± 27.82	5.351 ± 8.11	6.683 ± 9.68
T Mean(mm)	9.985 ± 19.28	12.509 ± 21.55	9.477 ± 11.85	10.958 ± 13.15
Roll(º)	0.575 ± 1.99	0.586 ± 2.00	0.566 ± 0.81	0.528 ± 0.75
Yaw(º)	0.845 ± 1.29	1.080 ± 1.52	0.868 ± 1.00	0.977 ± 1.14
Pitch(º)	1.372 ± 1.47	1.648 ± 1.77	1.380 ± 1.45	1.486 ± 1.57
R Mean(º)	0.931 ± 1.64	1.105 ± 1.83	0.938 ± 1.17	0.997 ± 1.26

Tabla 5.14 Comparativa del error de HPE entre usar el regresor MIX_1157_YP_1200 o el regresor MIX_3283_YP_1200.

Se puede observar que el uso de regresores entrenados con imágenes sintéticas que presentan altos valores de rotación roll (Tabla 5.13) proporciona menores valores de dispersión, lo que parece indicar una menor pérdida del regresor. Por otra parte, el uso de regresores entrenados con imágenes sintéticas que presentan altos valores de rotación yaw y pitch (Tabla 5.14), no evita la pérdida del regresor (altas desviaciones) pero sí que mejora el error medio obtenido.

5.5 Rotación de los frames + regresores entrenados

Si se combinan las dos soluciones implementadas, se obtienen los siguientes resultados:

	SSDR_1157		SSDR	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	9.266 ± 10.62	9.877 ± 9.59	10.322 ± 11.44	9.729 ± 9.14
Ty(mm)	13.771 ± 14.41	17.663 ± 16.69	13.978 ± 13.64	15.467 ± 13.86
Tz(mm)	5.892 ± 6.37	7.550 ± 9.00	5.340 ± 5.28	6.615 ± 7.23
T Mean(mm)	9.643 ± 11.44	11.697 ± 13.01	9.880 ± 11.29	10.604 ± 11.08
Roll(º)	0.527 ± 0.66	0.560 ± 0.68	0.622 ± 0.77	0.572 ± 0.66
Yaw(º)	0.759 ± 0.83	0.889 ± 0.91	0.885 ± 0.97	0.876 ± 0.83
Pitch(º)	1.401 ± 1.50	1.676 ± 1.69	1.423 ± 1.44	1.487 ± 1.49
R Mean(º)	0.896 ± 1.12	1.042 ± 1.26	0.977 ± 1.15	0.978 ± 1.12

Tabla 5.15 Comparativa del error de HPE usando el regresor SSDR_1157 o SSDR con rotación.

	MIX_1157_R_80		MIX_3283_R_140	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	8.855 ± 10.29	11.004 ± 10.87	9.638 ± 10.22	9.963 ± 9.37
Ty(mm)	14.566 ± 14.55	18.579 ± 17.25	13.408 ± 13.02	15.276 ± 13.71
Tz(mm)	5.865 ± 6.46	7.464 ± 8.91	4.733 ± 4.72	6.266 ± 6.92
T Mean(mm)	9.762 ± 11.53	12.349 ± 13.66	9.260 ± 10.55	10.502 ± 11.03
Roll(º)	0.472 ± 0.61	0.498 ± 0.62	0.634 ± 0.78	0.591 ± 0.67
Yaw(º)	0.741 ± 0.81	0.952 ± 0.98	0.830 ± 0.86	0.898 ± 0.84
Pitch(º)	1.477 ± 1.51	1.725 ± 1.72	1.366 ± 1.41	1.461 ± 1.46
R Mean(º)	0.897 ± 1.13	1.059 ± 1.30	0.943 ± 1.10	0.983 ± 1.11

Tabla 5.16 Comparativa del error de HPE usando el regresor MIX_1157_R_80 o MIX_3283_R_140 con rotación.

	MIX_1157_YP_1200		MIX_3283_YP_1200	
	BFM	**SFM**	**BFM**	**SFM**
Tx(mm)	8.661 ± 9.62	9.708 ± 10.02	8.969 ± 9.51	9.366 ± 9.11
Ty(mm)	12.748 ± 12.23	16.465 ± 15.04	13.263 ± 12.77	15.062 ± 13.02
Tz(mm)	5.293 ± 5.74	6.793 ± 8.04	4.784 ± 4.81	6.067 ± 6.57
T Mean(mm)	8.901 ± 10.05	10.988 ± 12.12	9.006 ± 10.21	10.165 ± 10.60
Roll(º)	0.462 ± 0.60	0.465 ± 0.60	0.565 ± 0.68	0.499 ± 0.55
Yaw(º)	0.698 ± 0.76	0.883 ± 0.95	0.766 ± 0.79	0.842 ± 0.81
Pitch(º)	1.316 ± 1.31	1.573 ± 1.57	1.359 ± 1.36	1.442 ± 1.39
R Mean(º)	0.826 ± 1.01	0.973 ± 1.20	0.897 ± 1.04	0.928 ± 1.06

Tabla 5.17 Comparativa del error de HPE usando el regresor MIX_1157_YP_1200 o MIX_3283_YP_1200 con rotación

Comparando las distintas tablas se observa que, al combinar ambas soluciones, los regresores entrenados con imágenes sintéticas que presentan altos valores de rotación yaw y pitch (Tabla 5.17) mejoran el error medio obtenido y no muestran altos valores de dispersión. Llama la atención que, pese a lo que pueda parecer en un principio, los regresores entrenados con un número menor de imágenes reales, funcionan mejor en todos los casos.

Además el uso del modelo BFM presenta mejores resultados que el uso del modelo SFM (sobre todo con el uso de regresores entrenados con imágenes sintéticas) por lo que, finalmente, se ha escogido éste como mejor modelo geométrico 3D.

De esta forma se puede concluir que los mejores resultados de HPE obtenidos mediante el software SSDM se consiguen es referenciando la pose respecto al primer frame (*zeroed avg*), utilizando la implementación de rotación de los frames y haciendo uso de la combinación regresor-modelo 3D:

- **Regresor:** *MIX_1157_YP_1200*.
- **Modelo 3D:** *BFM*.
- **Puntos del modelo 3D:** *Landmarks 3D*.

5.6 HPE propio del software

Como ya se ha comentado, tanto IF como SSDM tienen un método propio de estimación de la pose de la cabeza aunque proporcionan únicamente los valores de rotación.

En la Tabla 5.18 se muestran los resultados obtenidos por cada método propio. Los errores han sido calculados referenciando la pose respecto al primer frame (zeroed).

	IntraFace	SSDM
Roll(º)	0.896 ± 1.32	0.594 ± 0.65
Yaw(º)	3.116 ± 3.38	3.687 ± 3.90
Pitch(º)	2.201 ± 2.43	2.217 ± 2.53
R Mean(º)	2.071 ± 2.68	2.166 ± 2.99

Tabla 5.18 Resultados de error de HPE utilizando el método propio de IntraFace y SSDM.

Se puede ver que los errores obtenidos por estos métodos distan mucho de los obtenidos utilizando el algoritmo POSIT.

5.7 Resumen de resultados

Este apartado está dedicado a realizar la comparativa entre los errores de HPE obtenidos por los métodos originales, los métodos propios del software utilizado y el método optimizado para el software SSDM que presenta mejores resultados.

En la Tabla 5.19 se presentan los resultados obtenidos por cada uno de los métodos mencionados. Como ya se ha dicho, los métodos propios (*OWN*) únicamente proporcionan valores de estimación de rotación.

	IntraFace		SSDM		
	IF_126/BFM	**OWN**	**SSDR/BFM**	**OWN**	**MIX_1157_YP_1200/BFM**
Tx(mm)	9.659 ± 9.59	-	10.755 ± 12.41	-	**8.661 ± 9.62**
Ty(mm)	12.083 ± 12.16	-	14.016 ± 14.51	-	**12.748 ± 12.23**
Tz(mm)	5.665 ± 6.19	-	5.868 ± 8.66	-	**5.293 ± 5.74**
T Mean(mm)	9.136 ± 9.99	-	10.213 ± 12.56	-	**8.901 ± 10.05**
Roll(º)	0.383 ± 0.40	0.896 ± 1.32	0.613 ± 0.93	0.594 ± 0.65	**0.462 ± 0.60**
Yaw(º)	0.852 ± 0.84	3.116 ± 3.38	0.977 ± 1.08	3.687 ± 3.90	**0.698 ± 0.76**
Pitch(º)	1.216 ± 1.24	2.201 ± 2.43	1.425 ± 1.52	2.217 ± 2.53	**1.316 ± 1.31**
R Mean(º)	0.817 ± 0.96	2.071 ± 2.68	1.005 ± 1.25	2.166 ± 2.99	**0.826 ± 1.01**

Tabla 5.19 Comparativa entre los errores de HPE obtenidos por los métodos originales, los métodos propios del software utilizado y el método del software SSDM que presenta mejores resultados.

La primera conclusión que se obtiene al analizar la Tabla 5.19 es que la estimación de la pose obtenida por los algoritmos propios del software utilizado es peor que cualquiera de los métodos que hagan uso de POSIT.

En cuanto al mejor método de estimación de la pose, el uso de la combinación *IF_126/BFM* o el de *MIX_1157_YP_1200/BFM* proporcionan resultados similares tanto en error medio como en desviación típica. Sin embargo, se puede ver la mejora obtenida utilizando el software SSDM; inicialmente se tiene un error medio de 10.213 ± 12.56mm en traslación y 1.005 ± 1.25º en rotación (*SSDR/BFM*), y se consigue disminuir hasta 8.901 ± 10.05mm en traslación y 0.826 ± 1.01º en rotación.

La mejora obtenida utilizando el software SSDM ha sido posible gracias a que se trata de software de código abierto, lo que ha permitido realizar las modificaciones e implementaciones descritas en este trabajo. Por ello, aunque en números sea ligeramente superior la combinación *IntraFace/BFM*, se concluye que, debido a los buenos resultados y la alta capacidad de mejora, la mejor combinación a la hora de realizar la estimación de pose de la cabeza es *SSDM/BFM*.

Capítulo 6
Conclusiones y líneas futuras

En este capítulo se expone el conjunto de conclusiones obtenidas mediante la realización de este trabajo. También se dedica un apartado a comentar las futuras líneas de investigación que pueden surgir como consecuencia del mismo.

6.1 Conclusiones

Durante el análisis de resultados se ha visto la importancia de referenciar la pose respecto al primer frame y se ha comprobado que calcular la pose del primer frame a partir de la media de los landmarks obtenidos al repetir el primer frame 100 veces, afecta de manera positiva al HPE cuando se utiliza IntraFace como sistema de seguimiento facial. También se ha comprobado que no existe una diferencia aparente entre utilizar como puntos del modelo geométrico 3D, los Landmarks 3D o los Vértices 3D.

Con respecto a los modelos geométricos 3D, se ha visto que el uso de los modelos reconstruidos a partir del software 4DFace, no mejora la estimación respecto a utilizar el modelo medio SFM y, por consiguiente, no merece la pena el cálculo de los mismos. Por otro lado, el uso del BFM como modelo 3D, proporciona los mejores resultados en prácticamente la totalidad de los casos.

En cuanto a los sistemas de seguimiento facial, se ha visto que, en un principio, IntraFace proporcionaba mejores resultados que SSDM. Además, con los resultados obtenidos al utilizar la

base de datos sintética, se ha visto que el regresor SSDR tiene problemas a la hora de detectar los landmarks 2D en vídeos que presenten altos valores de rotación.

Con el uso de los regresores entrenados con imágenes sintéticas y del algoritmo de rotación en cada frame, se consigue suplir las limitaciones del software SSDM y obtener errores de HPE similares a los de IntraFace.

Por último y para dar respuesta al objetivo principal de este proyecto, se concluye que, debido a los buenos resultados y la alta capacidad de mejora, la mejor combinación a la hora de realizar la estimación de pose de la cabeza es utilizar; como sistema de seguimiento facial, el software *SSDM*; y como modelo geométrico 3D, el modelo medio *BFM*.

6.2 Líneas futuras

Las líneas de investigación que pueden surgir como consecuencia de los resultados obtenidos en este trabajo, pueden dividirse en 3 grandes grupos: Entrenamiento de regresores, reconstrucción de modelos particulares y optimización del sistema de seguimiento.

6.2.1 Entrenamiento de regresores

Al haberse desarrollado un entorno de trabajo con el cual poder comparar el HPE obtenido utilizando distintos sistemas de seguimiento y modelos 3D, se podría realizar un análisis más exhaustivo utilizando una mayor cantidad de regresores entrenados bajo distintas condiciones (cambios en la anchura W o la profundidad K del regresor).

Además, la adquisición de un ordenador de altas prestaciones por parte del Grupo de Investigación, abre la puerta a la realización de entrenamientos con un mayor número de imágenes e incluso realizando la inicialización a partir de los resultados obtenidos en el frame anterior. Otra implementación a la hora de entrenar nuevos regresores es la de no utilizar únicamente frames obtenidos de la base de datos sintética, sino generar imágenes realizando las traslaciones y rotaciones de las cabezas sintéticas que se consideren adecuadas.

6.2.2 Reconstrucción de modelos particulares

Otro campo de investigación es el relacionado con el uso de modelos sintéticos reconstruidos para cada usuario particular. Se ha visto que el uso de estos modelos no proporciona mejoras, sin embargo, puede ser interesante realizar un estudio que mida el error de reconstrucción de las cabezas sintéticas comparándolas con sus respectivos modelos groundtruth a fin de conseguir optimizar el proceso por el cual se generan estos modelos reconstruidos. Esto es posible ya que el software 4DFace, al igual que el SSDM es de código abierto y permite su manipulación.

6.2.3 Optimización del sistema de seguimiento.

En este grupo entrarían las implementaciones realizadas sobre el software SSDM a fin de mejorar su funcionamiento. Un ejemplo sería la implementación de un sistema de detección automática de pérdida del tracker basado en el cálculo y comparación de la pose de frames consecutivos.

Bibliografía

[1] R. Szeliski, Computer Vision: algorithms and applications, Springer Science & Business Media, 2010.

[2] L. G. Roberts, *Machine perception of three-dimensional soups,* Massachusetts Institute of Technology, 1963.

[3] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz y R. Szeliski, «Building rome in a day,» *Communications of the ACM,* vol. 54, nº 10, pp. 105-112, 2011.

[4] M. Ariz, J. J. Bengoechea, A. Villanueva y R. Cabeza, «A novel 2D/3D database with automatic face annotation for head tracking and pose estimation,» *Computer Vision and Image Understanding,* vol. 148, pp. 201-210, 2016.

[5] E. Murphy-Chutorian y M. M. Trivedi, «Head pose estimation in computer vision: A survey,» *IEEE transactions on pattern analysis and machine intelligence,* vol. 31, nº 4, pp. 607-626, 2009.

[6] P. Viola y M. J. Jones, «Robust real-time face detection,» *International journal of computer vision,* vol. 57, nº 2, pp. 137-154, 2004.

[7] X. Xiong y F. De la Torre, «Supervised descent method for solving nonlinear least squares problems in computer vision,» *Computing Research Repository,* 2014.

[8] D. F. Dementhon y L. S. Davis, «Model-based object pose in 25 lines of code,» *International journal of computer vision,* vol. 15, nº 1-2, pp. 123-141, 1995.

[9] R. Segura, R. Cabeza y M. Ariz, «Reconstrucción 3D de modelos personalizados de cabeza para la estimación de la pose,» *Public University of Navarre,* 2015.

[10] P. Paysan, R. Knothe, B. Amberg, S. Romdhani y T. Vetter, «A 3D face model for pose and illumination invariant face recognition,» *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on,* pp. 296-301, 2009.

[11] M. Ariz, Contributions to Head Pose Estimation Methods, Pamplona: Public University of Navarre, 2016.

[12] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou y M. Pantic, «A semi-automatic methodology for facial landmark annotation,» *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops,* pp. 896-903, 2013.

[13] M. Köstinger, P. Wohlhart, P. M. Roth y H. Bischof, «Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization,» *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference,* pp. 2144-2151, 2011.

[14] V. Le, J. Brandt, Z. Lin, L. Bourdev y T. Huang, « Interactive facial feature localization,» *Computer Vision–ECCV 2012,* pp. 679-692, 2012.

[15] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman y N. Kumar, «Localizing parts of faces using a consensus of exemplars,» *IEEE transactions on pattern analysis and machine intelligence,* vol. 35, nº 12, pp. 2930-2940, 2013.

[16] D. G. Lowe, «Distinctive image features from scale-invariant keypoints,» *International journal of computer vision,* vol. 60, nº 2, pp. 91-110, 2004.

[17] N. Dalal y B. Triggs, «Histograms of oriented gradients for human detection,» *Computer Vision and Pattern Recognition, CVPR 2005,* vol. 1, pp. 886-893, 2005.

[18] F. De la Torre, W. S. Chu, X. Xiong, F. Vicente, X. Ding y J. Cohn, «Intraface,» *Automatic Face and Gesture Recognition (FG),* vol. 1, pp. 1-8, 2015.

[19] Z. Feng, H. P. H., J. Kittler, W. Christmas y X. J. Wu, «Random cascaded-regression copse for robust facial landmark detection,» *IEEE Signal Processing Letters,* vol. 22, nº 1, pp. 76-80, 2015.

[20] P. Huber, G. Hu, R. Tena, P. Mortazavian, W. P. Koppen, W. Christmas, M. Ratsch y J. Kittler, «A multiresolution 3D Morphable Face Model and fitting framework,» *International Conference on Computer Vision Theory and Applications,* pp. 1-8, 2016.

[21] P. Huber, W. Christmas, A. Hilton, J. Kittler y M. Rätsch, «Real-time 3D face super-resolution from monocular in-the-wild videos,» *ACM SIGGRAPH 2016 Posters,* p. 67, 2016.