E.T.S. DE INGENIERÍA INDUSTRIAL, INFORMÁTICA Y DE TELECOMUNICACIÓN

Caracterización de la reconstrucción tridimensional mediante técnicas multivista

Grado en Ingeniería en Tecnologías de Telecomunicación

Trabajo Fin de Grado

DAN LÓPEZ PUIGDOLLERS
RAFAEL CABEZA LAGUNA
ARANTXA VILLANUEVA LARRE

Pamplona, 1 de Julio de 2016

Para mis padres, por vuestro apoyo incondicional.
Gracias a mi hermana María, por inspirarme con tu ejemplo.
Gracias a mis tutores Rafael y Arantxa, por vuestra disposición y ayuda.

"Algo que estábamos reteniendo nos debilitaba, hasta que descubrimos que éramos nosotros mismos." Robert Frost (1874-1963)

Índice

Resumen

Este trabajo versa sobre un conjunto de técnicas de estimación de estructuras tridimensionales (SfM, *Structure from Motion*), situadas dentro del campo de visión artificial. Son propicias para la reconstrucción de modelos tridimensionales, a partir de la adquisición de un conjunto de imágenes. Tienen la principal ventaja de ser poco invasivas y de bajo coste. El fin es crear una nueva implementación de las técnicas desarrolladas por Rubén Segura Aréjula en este ámbito, tanto para modelos de cabeza generados sintéticamente como para modelos reales, a partir de los nuevos algoritmos y funciones de MATLAB 2016a en materia de extracción, detección y seguimiento de características locales de una imagen; estimación de movimiento y reconstrucción en 3D, evaluando la eficacia del nuevo conjunto de funciones de la nueva librería para el desarrollo de diversos métodos de reconstrucción tridimensional.

Palabras clave:
Structure from Motion - SfM – MATLAB – IntraFace – Basel Face Model – *tracking* KLT

Abstract

This work deals with a set of techniques for estimating three-dimensional structures (SfM, Structure from Motion), located within the field of artificial vision. They are conducive to the reconstruction of three-dimensional models, from the acquisition of a set of images. They have the main advantage of being less invasive and low cost. The aim is to create a new implementation of the techniques developed by Ruben Segura Aréjula in this area, both for head models generated synthetically as real models, from new algorithms and functions of MATLAB 2016a on extraction, detection and tracking local characteristics of an image; motion estimation and three-dimensional reconstruction, evaluating the effectiveness of the new set of functions of the new library for the development of various methods of three-dimensional reconstruction.

Keywords:
Structure from Motion - SfM – MATLAB – IntraFace – Basel Face Model – tracking KLT

1. Introducción.

1.1. Perspectiva actual de la visión artificial y objetivos.

En la última década, los avances en el campo de visión artificial han sido notables, ligados estrechamente a la evolución de los sistemas de adquisición de imágenes y a la exponencial mejora de procesamiento de los sistemas computacionales actuales, razón por la cual nos ha permitido obtener diversos métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratados en distintas disciplinas y en múltiples aplicaciones.

Como ejemplos de sistemas incluidos en aplicaciones de visión artificial podrían citarse los siguientes:

- Control de los procesos, por ejemplo, un robot industrial.
- De navegación, por ejemplo, mediante un vehículo autónomo o robot móvil.
- La detección de eventos, por ejemplo, para la vigilancia visual o el conteo de personas.
- La organización de la información, por ejemplo, para indexar bases de datos de imágenes y secuencias de imágenes.
- Modelado de objetos o entornos, por ejemplo, análisis de imágenes médicas o de modelado topográfico.
- La interacción, por ejemplo, como la entrada a un dispositivo para la interacción hombre-máquina.
- Inspección automática, por ejemplo, en aplicaciones de fabricación.

Este proyecto versa sobre un conjunto de técnicas situadas en el campo de visión artificial propicios para la reconstrucción de modelos tridimensionales a partir de la adquisición de un conjunto de imágenes. Tienen la principal ventaja de ser poco invasivas y de bajo coste.

El fin es crear una nueva implementación de las técnicas desarrolladas en el TFG de Rubén Segura Aréjula [1] por medio de los nuevos algoritmos y funciones de MATLAB 2016a en materia de extracción, detección y seguimiento de características de una imagen, estimación de movimiento y reconstrucción en 3D, un conjunto incluido en Computer Vision System Toolbox™.

Se evaluará la eficacia del nuevo conjunto de funciones de la nueva librería un método de reconstrucción tridimensional denominado *Structure from Motion* (SfM), así como el planteamiento de diversos algoritmos que incluyan estas funciones.

1.2. Estado del arte.

1.2.1. Structure from Motion.

Structure from Motion (SfM) es el proceso de estimación de la estructura tridimensional de una escena a partir de un conjunto de imágenes bidimensionales. SfM se utiliza en muchas aplicaciones, tales como el escaneo 3D y la realidad aumentada. Se introduce dentro de las técnicas conocidas como *range imaging*: triangulación en estéreo, luz estructurada, ToF (*Time of Flight*), o Apertura Codificada.

SfM se puede calcular de muchas maneras diferentes. Depende de diferentes factores, como el número y tipo de cámaras utilizadas, y si las imágenes están ordenadas. Para calcular la estructura y el movimiento en unidades del mundo (métricas), se necesita información adicional:

- El tamaño de un objeto en la escena.
- Información de otro sensor, por ejemplo, un odómetro.

SfM se puede aplicar a partir de dos vistas, pero para la mayoría de aplicaciones, como la robótica y la conducción autónoma, SfM utiliza más de dos puntos de vista.

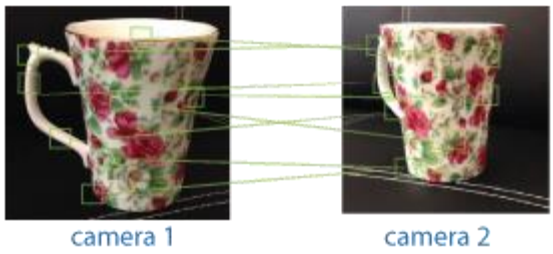


Figura 1.1. Dos vistas de un objeto y correspondencia de los puntos comunes entre las respectivas imágenes.

La filosofía empleada en SfM a partir de dos puntos de vista se puede extender para varios puntos de vista. El conjunto de múltiples puntos de vista utilizado por SfM puede estar ordenado o desordenado. Se asume una secuencia ordenada de puntos de vista. SfM desde múltiples puntos de vista requiere correspondencias de puntos en varias imágenes, llamadas *tracks*.

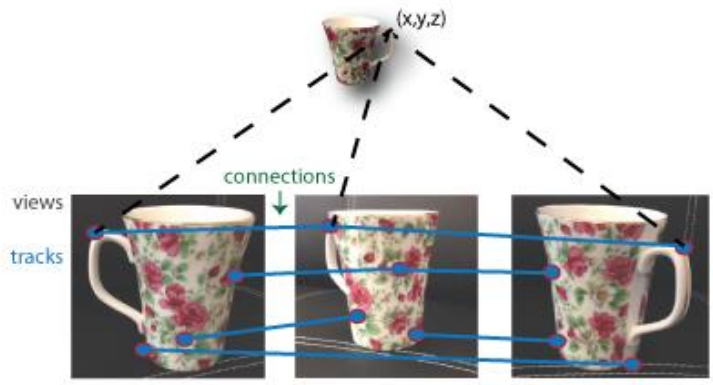


Figura 1.2. Múltiples vistas de un objeto y correspondencia de los puntos comunes entre las respectivas imágenes.

Similar al procedimiento empleado en SfM a partir de dos puntos de vista, se puede encontrar la pose de la cámara 2 con respecto a la cámara 1. Para aplicar este enfoque para el caso de varias vistas, se encuentra la pose de la cámara 3 respecto a la cámara 2, y así sucesivamente. Las poses relativas deben

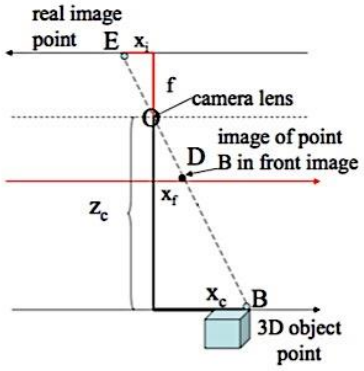
transformarse en un sistema de coordenadas común. Típicamente, todas las poses de la cámara se calculan con relación a la cámara 1 de manera que todas las poses están en el mismo sistema de coordenadas.

Cada estimación de la pose de la cámara de una vista a la siguiente contiene errores. Los errores se deben a la localización de puntos imprecisos en imágenes y/o calibración imprecisa. Estos errores se acumulan conforme el número de vistas aumenta, un efecto conocido como *drift* o deriva. Una forma de reducir la deriva, es refinar poses de cámara y ubicaciones de los puntos 3D.

Una vez extraídos los puntos característicos y sus respectivos *tracks*, podemos hallar la posición de la cámara, en términos de rotación y traslación, ya sea por medio del cálculo de la matriz fundamental o algoritmos de estimación de pose, a través de dichos puntos, si conocemos los parámetros intrínsecos de la cámara, si se ha realizado la calibración previa de la misma.

Los parámetros de una cámara permiten generar una relación entre el sistema de coordenadas del sensor expresado en píxeles respecto al sistema de coordenadas del mundo en unidades métricas. El proceso de calibración de la cámara permite obtener un parámetro esencial en la transformación de coordenadas, como es la distancia o longitud focal, *f*, así como las coordenadas del centro óptico, asumiendo en todo momento un sistema *pinhole*. A su vez, el proceso de calibración también nos permite calcular parámetros relacionados con la distorsión radial y tangencial que genera la lente de la cámara. Puede resultar muy interesante si se quiere minimizar el error de proyección de los puntos reduciendo el efecto de deriva.

La matriz intrínseca es empleada en todo el proceso de reconstrucción SfM, porque es imprescindible conocer la relación de transformación aplicada a los puntos.



$$\frac{x_i}{f} = \frac{x_c}{z_c}$$

Figura 1.3. Esquema básico de correspondencia entre coordenadas del mundo de un punto (B) y su equivalente proyección en el plano de imagen de la cámara (E).

Estos parámetros se engloban dentro de una matriz 3x3, denominada matriz intrínseca.

$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

Ecuación 1.1. Matriz intrínseca de una cámara.

Las coordenadas $[c_x\ c_y]$ representan el centro óptico (el punto principal), en píxeles.

$f_x = F \cdot s_x$.
$f_y = F \cdot s_y$.

Donde F, es la longitud focal en unidades del mundo (unidad métrica), normalmente expresada en milímetros.

$[s_x, s_y]$ son el número de píxeles por unidad del mundo en la dirección *X* e *Y*, respectivamente.
f_x y f_y se expresan en píxeles.

Al trabajar con matrices, existe la posibilidad de hacerlo de manera más compacta empleando un sistema de coordenadas homogéneas. Consiste en añadir una componente más tanto a las coordenadas de la imagen (coordenadas 2D), como a las coordenadas de la escena (coordenadas 3D).

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous image coordinates **homogeneous scene coordinates**

Ecuación 1.2. Transformación a coordenadas homogéneas.

Realizada la transformación al nuevo sistema de coordenadas, los parámetros permiten establecer la relación de proyección de los puntos en el espacio en el plano imagen de acuerdo a la siguiente fórmula:

$$\begin{bmatrix} s_x \\ s_y \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Ecuación 1.3. Transformación en coordenadas homogéneas de los puntos de imagen en coordenadas del mundo, donde s se refiere al factor de escala [2].

Existen varias formas de realizar la calibración, aunque la más empleada es empleando una superficie patrón llamada *checkerboard*, donde se localizan un determinado número de puntos asociados con las esquinas en un gran número de imágenes distintas, entre 10 y 20, colocando preferentemente el tablero en distintas posiciones. Se realiza la calibración y se comprueba su precisión. Después se ajustan los parámetros, si es necesario.

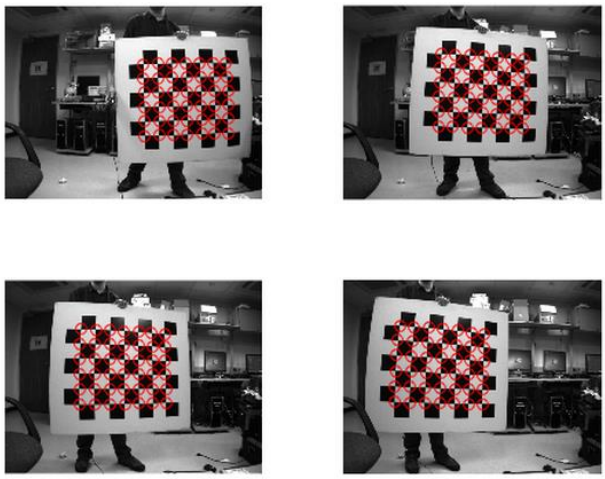


Figura 1.4. Ejemplo de calibración de una cámara. Detección de puntos de un *checkerboard*, generalmente un damero, desde distintas vistas.

De forma adicional, se puede extraer la matriz extrínseca, que localiza la cámara dentro de las coordenadas del mundo, y en qué dirección está apuntando. Se define por un vector R de rotación y t de

traslación, aunque se obtiene más a menudo por medio del cálculo de las poses de la cámara mediante otros procedimientos como la Matriz Fundamental.

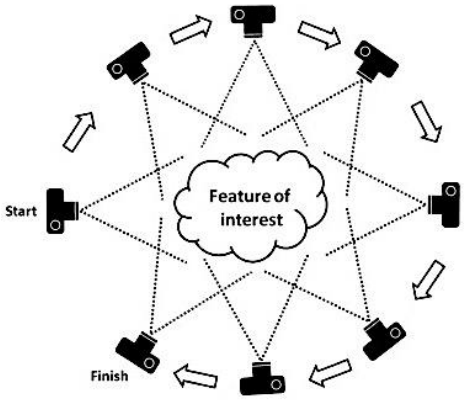


Figura 1.5. SfM. En lugar de un único par estéreo, la técnica SfM requiere múltiples fotografías, superpuestas como entrada a la extracción de características y algoritmos de reconstrucción [3].

La posición relativa de la cámara respecto al objeto o escena, la correspondencia de puntos característicos en imágenes sucesivas, *tracks*, y los parámetros intrínsecos de la cámara nos permiten realizar un proceso de triangulación para obtener la ubicación 3D de los puntos.

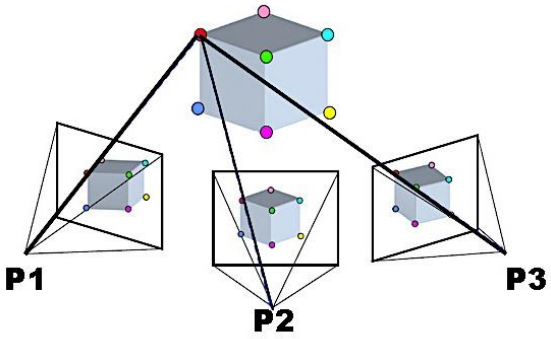


Figura 1.6. Extracción de características comunes de tres imágenes distintas para realizar la triangulación.

El algoritmo de optimización no lineal, llamado *Bundle Adjustment*, una variante del algoritmo de Levenberg-Marquardt, se utiliza casi siempre como el último paso del algoritmo de reconstrucción. Equivale a un problema de optimización de la estructura y parámetros de visualización (es decir, poses de las cámaras, calibración intrínseca y distorsión radial), para obtener una reconstrucción que es óptima bajo ciertas hipótesis respecto al ruido correspondiente a las características de las imágenes observadas, es decir, si el ruido se puede caracterizar como gaussiano de media cero, el algoritmo trabaja de manera óptima.

SfM se estudia y se implementa en los campos de visión artificial y de percepción visual. En el campo de visión biológica, SfM se relaciona con el fenómeno que tienen los humanos y otros seres vivos de generar la percepción 3D de una escena a partir de los puntos 2D proyectados en la retina con diferencias de movimiento o localización.

1.2.2. Extracción y detección de características locales.

Para tener correspondencias entre puntos, se deben encontrar entre las sucesivas imágenes un conjunto de puntos característicos comunes e invariantes que cumplan ciertas propiedades por medio de algoritmos que detectan ciertas características locales.

Las características locales se refieren a un patrón o estructura distintiva que se encuentra en una imagen, tal como un punto, un borde o una región. Por lo general se asocian con una región de la imagen que difiere de su entorno inmediato por la textura, el color o intensidad. Lo que representa la característica en realidad no importa, sólo que es diferente de su entorno. Ejemplos de características locales son manchas, esquinas y los píxeles del borde.

Los detectores que se basan en gradiente y basados en la variación de intensidad alcanzan a detectar buenas características locales. Estas características incluyen bordes, manchas y regiones. Buenas características locales presentan las siguientes propiedades:

- **Detecciones repetibles**: Cuando se dan dos imágenes de una misma escena, la mayoría de las características que el detector encuentre en ambas imágenes son las mismas. Las características son robustas a cambios en las condiciones de visión y de ruido.
- **Distintivo:** La zona alrededor del centro de la característica varía lo suficiente como para permitir una comparación fiable entre las características.
- **Localizable**: La característica tiene un lugar único asignado a la misma. Los cambios en las condiciones de visión no afectan a su ubicación.

La detección de características selecciona regiones de una imagen que tienen un contenido único, como esquinas o manchas, para encontrar puntos de interés que se pueden utilizar para un futuro procesado. Estos puntos no se corresponden necesariamente a estructuras físicas, como las esquinas de una mesa. La clave para la detección de características es encontrar características que permanecen invariables a nivel local para que se puedan detectar incluso en presencia de rotación o cambio de escala.

La extracción de características implica el cálculo de un descriptor, que normalmente se realiza en las regiones centradas alrededor de características detectadas. Los descriptores se basan en el procesamiento de imágenes para transformar una región local de píxeles en una representación vectorial compacta. Esta nueva representación permite la comparación entre las regiones adyacentes, independientemente de los cambios en la escala o la orientación. Descriptores, tales como SIFT o SURF, se basan en cálculos de gradiente locales. Descriptores binarios, como BRISK o FREAK, dependen de los pares en las diferencias de intensidad locales, que luego son codificados en un vector binario.

Si los cambios entre imágenes son pequeños, es posible localizar los puntos en las sucesivas vistas empleando un algoritmo de *tracking* o seguimiento. Para el seguimiento de éstos es común el empleo del algoritmo de Kanade-Lucas-Tomasi, que hace posible el registro entre los sucesivos fotogramas y añade la ventaja de ser un método eficiente dentro de los métodos de búsqueda y seguimiento de características.

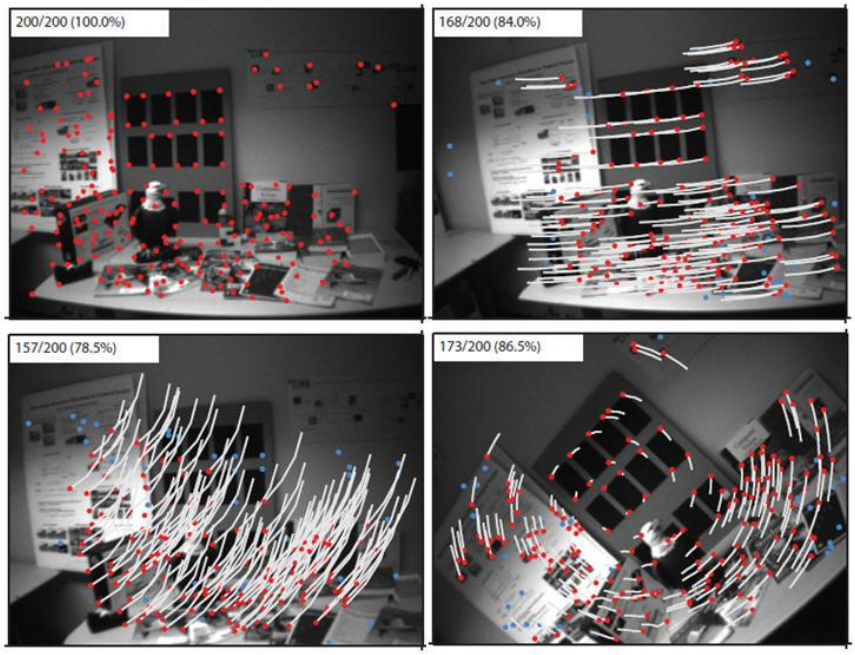


Figura 1.7. Ejemplo de funcionamiento del algoritmo de Kanade-Lucas-Tomasi ante movimiento en una cámara.

La eficiencia y robustez de cada algoritmo está ligada al tipo de aplicación, pues como se ha mencionado anteriormente, cada algoritmo detecta características locales distintas. Por ejemplo, SURF encuentra características de tipo *blob*, mientras BRISK lo hace con características de tipo *corner,* con sus respectivas implementaciones. Existen múltiples métodos de detección de características locales. A continuación, se citan y se describen de forma breve los más empleados:

- **HoG (*Histogram of Oriented Gradients*):** la forma o contorno de un objeto puede describirse como una distribución de gradientes de intensidad.
- **LBP (*Local Binary Patterns*):** es un descriptor usado para clasificación de texturas que, combinado con el descriptor HoG, mejora considerablemente el rendimiento en la detección en algunos casos.
- **SURF (*Speeded Up Robust Features)*:** es un detector y un descriptor de alto rendimiento de puntos característicos de una imagen, donde se transforma la imagen en coordenadas, utilizando una técnica llamada multi-resolución [4].

  Sus principales características son:

  - Invariante a escala y rotación (SURF vertical, solo invariante a escala).
  - Cálculo rápido.
  - Obtenido a partir de imágenes integrales y un vector de características de menor tamaño.
  - Ideas de SIFT, pero más simplificado.

  La detección consiste en hacer una réplica de la imagen original de forma Piramidal Gaussiana o Piramidal Laplaciana, y obtener imágenes del mismo tamaño, pero con un ancho de banda reducido. De esta manera se consigue un efecto de borrosidad sobre la imagen original, llamado scale space pyramid. Esta técnica asegura que los puntos de interés son invariantes en el escalado. El algoritmo SURF está basado en el predecesor SIFT.

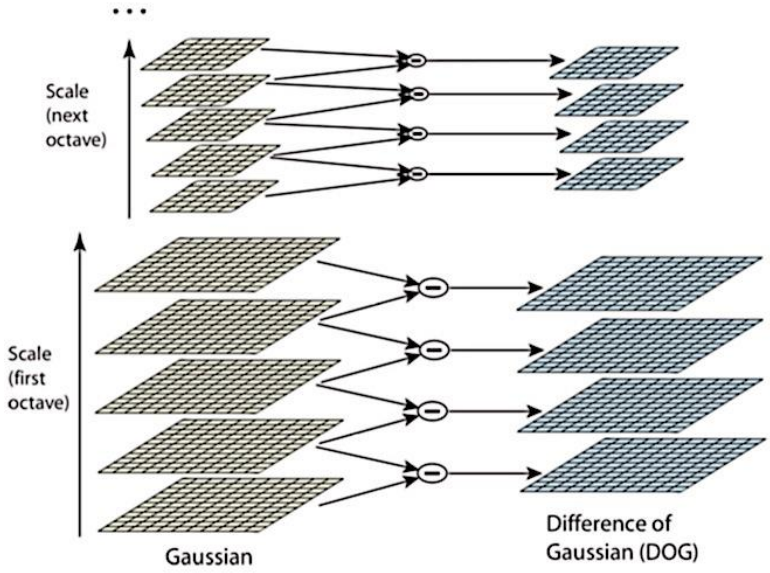


Figura 1.8. Ejemplo de *scale space pyramid* empleado por SURF [5].

La descripción se basa en la distribución de intensidad del contenido dentro del punto de interés de los puntos vecinos. Es generado basándose en el área circundante de un punto de interés, por lo que, realmente, se obtiene un vector descriptor para cada punto de interés.

- **FREAK (*Fast Retina Keypoint):*** alternativa del descriptor binario BRISK, cuya descripción se realiza comparando intensidades por medio de un patrón de muestreo circular, de manera más rápida y eficiente, empleando menos memoria.
- **BRISK (*Binary Robust Invariant Scalable Keypoints*):** es un detector tipo *corner* y descriptor binario que tiene la principal ventaja de tener menor complejidad computacional. La construcción de los vectores se hace por medio de comparaciones de intensidad, tomando una pareja de puntos muestreados y suavizados. Si el primer punto tiene mayor intensidad, la salida es un bit a uno. En otro caso, la salida es cero. Al ser un descriptor binario, permite que sea más rápido que SIFT/SURF. En los vectores se emplea distancia de Hamming, en vez de distancia euclídea.

  Este descriptor está especialmente orientado para requerimientos en tiempo real y dispositivos de baja potencia.

  La descripción de los puntos clave se realiza por medio de comparativas simples de brillo y muestreo de las regiones vecinas, de forma que son deterministas, equidistantes y circularmente concéntricas. El patrón de muestras de puntos de BRISK está formada por anillos concéntricos. Considerando una muestra de un punto, se toma una pequeña región alrededor y se le aplica un filtrado gaussiano. Los círculos rojos en la figura siguiente muestran el tamaño de la desviación estándar del filtro gaussiano para cada punto.

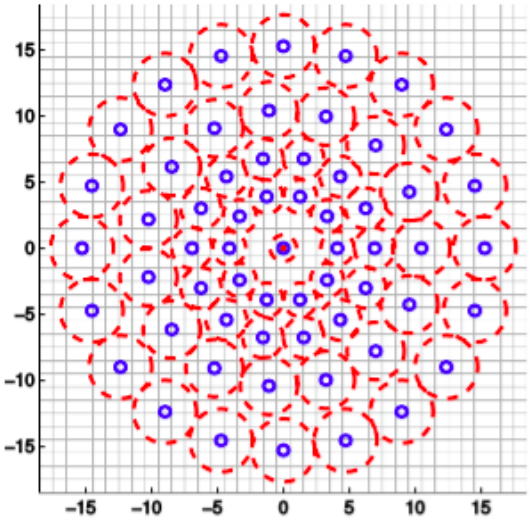


Figura 1.9. Descripción de puntos en BRISK [6].

- **Block (vecindario cuadrático simple):** método sencillo que consiste en extraer las regiones vecinas totalmente contenidas en los límites de la imagen.

La siguiente tabla nos define las propiedades más representativas de los diferentes tipos de descriptores ya descritos, con sus respectivas ventajas y aplicaciones. Para la reconstrucción tridimensional, lo más común es hacer uso de descriptores que sean invariantes a escala y rotación, es decir, una localización de puntos robusta frente a cambios en las poses de la cámara entre sucesivas imágenes, así como la posibilidad de hallar correspondencia de puntos.

Los descriptores que cumplen dichas características son SURF, FREAK y BRISK.

Descriptor	Binario	Invarianza		Uso típico	
		Escala	Rotación	Encontrar correspondencia entre puntos	Clasificación
HOG	No	No	No	No	Sí
LBP	No	No	Sí	No	Sí
SURF	No	Sí	Sí	Sí	Sí
FREAK	Sí	Sí	Sí	Sí	No
BRISK	Sí	Sí	Sí	Sí	No
Block	No	No	No	Sí	Sí

Tabla 1.1. Comparativa entre distintos algoritmos de descripción de características locales.

Tanto SURF como BRISK emplean técnicas de detección distintas, aunque ambas cumplen un criterio de invarianza a escala y rotación, de forma que se detecta una cierta cantidad de puntos con un criterio de patrón propio.

Figura 1.10. A la izquierda, detección de puntos SURF en una imagen. A la derecha, detección de puntos BRISK.

A continuación, se muestra la comparativa entre estos descriptores y las distintas combinaciones en su uso como detectores en términos de rendimiento computacional, invariancia a escala y a rotación [5].

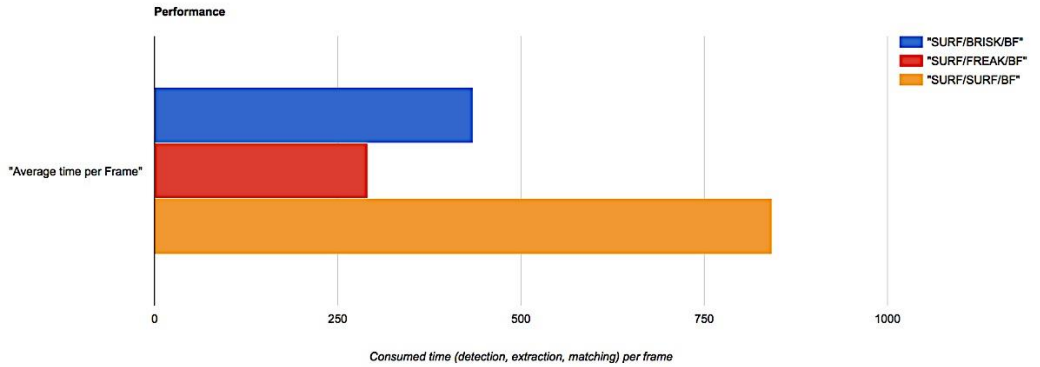


Figura 1.11. Comparativa de tiempos de ejecución de detección con SURF y descripción con SURF, FREAK o BRISK. Cuanto menor sea el resultado, mejor.

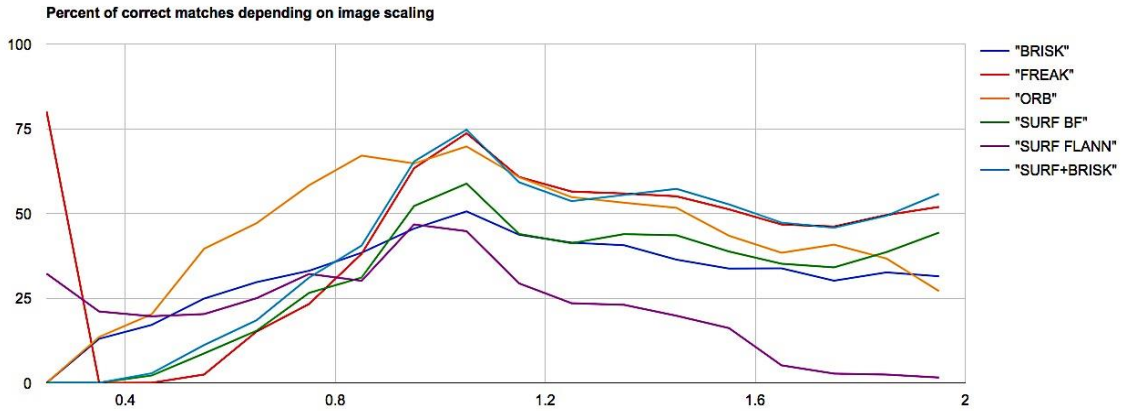


Figura 1.12. Comparativa de porcentaje de puntos correctos encontrados frente a cambios de escala. Mayor porcentaje, mejor.

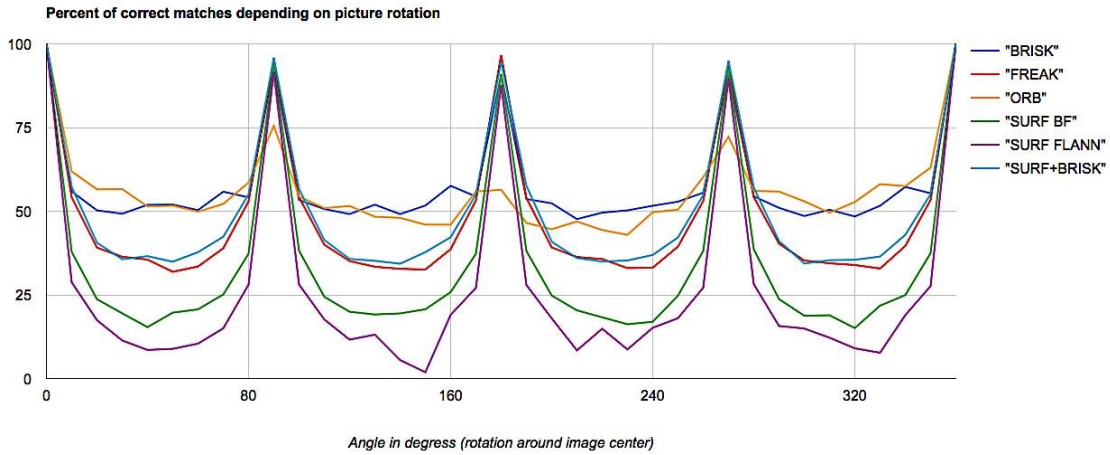


Figura 1.13. Comparativa de porcentaje de puntos correctos encontrados frente a cambios de rotación en grados. Mayor porcentaje, mejor.

El comportamiento de los tres algoritmos es similar, aunque cabe destacar la eficiencia del algoritmo BRISK en tiempos de ejecución, al tratarse de un descriptor binario. Observamos que se obtienen resultados muy buenos en el porcentaje de puntos correctos en la modificación del escalado o rotación de la imagen si se combina con detección SURF.

En general, los descriptores son más sensibles a los cambios de escala que a los cambios de rotación, como queda patente en las gráficas anteriores. La correspondencia de puntos es menos robusta cuando partimos de una imagen a otra con una escala menor, debido a la implementación que tienen estos descriptores de submuestrear la imagen para calcular los puntos mediante herramientas de sucesivos filtrados como la Piramidal Laplaciana, empleando distintas octavas con distintos tamaños de máscara de filtrado. Si ampliamos la escala, el comportamiento en el encuentro de correspondencias es más estable.

2. Reconstrucción tridimensional de un modelo de cabeza mediante puntos de *IntraFace*. Estimación de las poses de la cámara a partir del modelo medio de BFM.

2.1. Bases de datos: sintética y real.

2.1.1. Descripción de las bases de datos.

El primer paso consiste en elaborar un procedimiento que nos permita validar una correcta reconstrucción de un modelo de cabeza haciendo uso de estas funciones. Para ello, es necesario emplear un conjunto de modelos de cabeza completamente definidos, es decir, que conozcamos en todo momento sus características necesarias de manera exacta para la reconstrucción 3D para tener una referencia y poder calcular el error absoluto que estamos teniendo escogiendo un procedimiento o de las mismas funciones de MATLAB.

Estas características forman parte de nuestro *ground truth* y pueden ser tanto las poses de la cámara en cualquier fotograma de los vídeos, puntos característicos de la imagen, parámetros intrínsecos de la cámara y el modelo 3D real.

Disponemos de una base de datos de modelos de cabeza sintética que empleó Rubén Segura Aréjula en su TFG [1], 10 modelos de cabeza generados a partir de un modelo deformable de cabeza denominado Basel Face Model (BFM), que a su vez nos permite generar vídeos proyectando la cabeza en un plano realizando una transformación en traslación y rotación. Todo ello se hizo con el simulador desarrollado por José Javier Bengoechea [9] a partir del Proyecto de Fin de Carrera de Cristian Ordoyo [8].

La base de datos está formada por 10 usuarios distintos: 5 extraídos directamente de las cabezas ya creadas en la base de datos de BFM, y otros 5 generados de manera aleatoria. Para cada usuario, mediante el simulador, se han creado 12 vídeos con movimientos en traslación (componentes: [x y z]) y en rotación (componentes: [*roll yaw pitch*]). Es importante seguir un convenio en los sistemas de coordenadas de la cámara y de la cabeza, pues la proyección de los puntos de un modelo 3D en el sistema de la cámara debe ser unívoco.

En la rotación de la cabeza, trabajamos con los ángulos clásicos de Euler o Tait-Bryan, el mismo sistema empleado en la navegación, llamados eje de guiñada (*yaw* en inglés), de cabeceo (*pitch*) y de alabeo (*roll*).

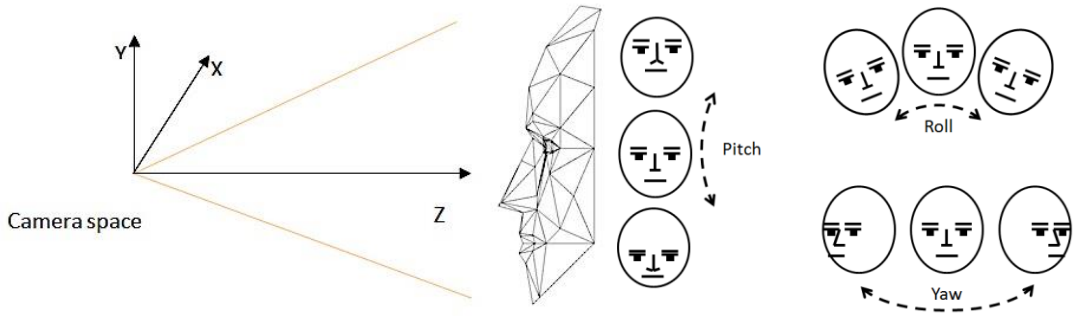


Figura 2.1. Sistema de coordenadas de la cámara y ejes de rotación de la cabeza.

En MATLAB, la forma en la que se opera mediante ángulos de rotación es por medio de una matriz de rotación 3x3. Existe un método para realizar el cambio de base de un vector de tres componentes a su matriz equivalente, y otro para hacer la conversión inversa.

Para realizar la primera conversión, se realiza el siguiente procedimiento. Dados unos ángulos de Euler θ_x, θ_y, θ_z:

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix}$$

$$Y = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix}$$

$$Z = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = ZYX$$

Ecuación 2.1. Conversión de ángulos de Euler a matriz de rotación.

Donde *R* representa la matriz de rotación 3x3.

Para realizar la conversión de matriz de rotación 3x3 a vector de ángulos de Euler, se realiza el siguiente procedimiento. Dada una matriz,

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Ecuación 2.2. Descomposición de la matriz de rotación.

Los ángulos se calculan como,

$$\theta_x = atan2(r_{32}, r_{33})$$
$$\theta_y = atan2(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2})$$
$$\theta_z = atan2(r_{21}, r_{11})$$

Ecuación 2.3. Conversión de matriz de rotación a ángulos de Euler.

Donde atan2 hace referencia a la función arcotangente en notación de C o MATLAB. Es importante que el convenio empleado en las unidades de los ángulos sean radianes para poder aplicar estas ecuaciones de descomposición en ambos casos.

2.1.2. Descripción y uso de la base de datos sintética.

Cada usuario está conformado por un total de 12 vídeos. Los vídeos del 1 al 6 tienen movimientos en una única componente, ya sea de traslación o rotación. Orientado a la reconstrucción con SfM, nos centraremos en los vídeos de rotación pura en *pitch* y en *yaw*, pues los cambios de traslación no afectan a la triangulación, pues en estos casos la cabeza únicamente se está desplazando por el plano imagen. El movimiento en *roll* tampoco aporta información nueva, ocurre lo mismo que en el caso de traslación.

Los vídeos del 1 al 3 son vídeos con movimiento en traslación *x, y* y *z*, respectivamente. El vídeo 4 se relaciona con un movimiento tipo *roll*.

Se escogen los vídeos 5 y 6, movimientos puros en *yaw* y *pitch*, respectivamente. Los vídeos del 7 al 12 son movimientos arbitrarios, combinación de movimiento en cualquier componente de traslación y rotación.

Los videos tienen una duración de 10 segundos a una tasa de 30 fps, lo que comprende un total de 300 fotogramas por vídeo. La resolución empleada es de 1280 x 720 pixeles. Los vídeos están contenidos en los archivos *user_xx_video_yy.avi*, donde *xx* se corresponde al número de usuario y *yy*, al número del vídeo. De estos archivos se extraen los fotogramas.

Todos los vídeos de cada usuario tienen referenciados además ficheros de datos con información acerca de la localización de los 43 puntos faciales de *IntraFace* [7], vectores de traslación y rotación que se han aplicado a la cabeza en el origen para moverlo a la posición que adoptan en cualquier fotograma, los cuales nos indican la posición de la cabeza o en inglés *Head Position*; parámetros intrínsecos de la cámara y el modelo 3D ideal de la cabeza.

Generalmente, MATLAB suele ser específico en el formato de los parámetros de entrada de cada función. Debido a esto, es necesario conocer la información de la base de datos y en qué manera se presenta para poder adaptarla al formato común de MATLAB.

Los 43 puntos detectados con *IntraFace* para cada fotograma con y sin promediado están almacenados en un archivo, siguiendo el formato *user_xx_video_yy_intraface.mat*, para los puntos promediados en 20 iteraciones y *user_xx_video_yy_intraface2.mat*, para los puntos sin promediar, donde *xx* se corresponde al número de usuario y *yy*, al número del vídeo.

Para cada usuario y para cada vídeo, existe un archivo *user_xx_video_yy_results.mat*, fichero que aúna el modelo 3D original, todos los parámetros de la cámara, *ground truth* y los puntos de *IntraFace*: ideales (*ground truth* 2D), promediados y sin promediar. También se incluye en una matriz el movimiento realizado en rotación y traslación para llevar a la cabeza a la posición de cada fotograma, que también forma parte del *ground truth*. Este fichero se divide en tres estructuras que contemplan separadamente todos los parámetros citados anteriormente: *Model*, *Results* y *Simulation*.

Model representa lo que se conoce como el *ground truth* 3D, el modelo de cabeza tridimensional, con los 53490 puntos con los que se generó el modelo y también los 43 puntos de *IntraFace* indexados a este modelo tridimensional denso. También aparecen los coeficientes que se emplearon para generar el modelo con BFM, por lo que es posible replicar de nuevo el modelo si se requiere en otra ocasión.

Results incluye el *ground truth* 2D y los puntos de *IntraFace* promediados y sin promediar.

Simulation incluye también el modelo, pero la información relevante de esta estructura son los parámetros de la cámara empleados en la simulación, información muy importante que está presente en numerosas etapas de la reconstrucción mediante SfM. *Motion* representa el *ground truth* de las poses.

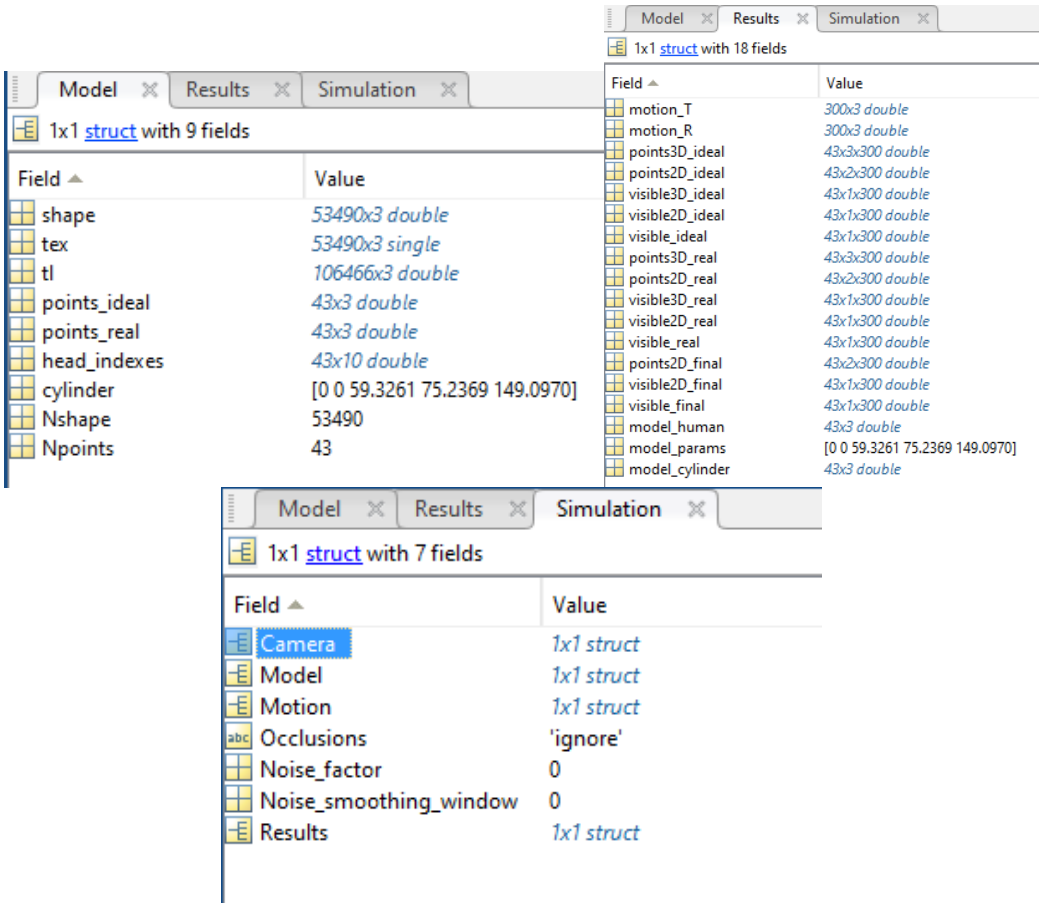


Figura 2.2. Contenido de un fichero *user_xx_video_yy_results.mat.*

2.1.3. Descripción y uso de la base de datos real.

La base de datos real replica el contenido y prácticamente el formato que se emplea en la base de datos sintética, salvo algunos detalles como la resolución de los vídeos, que en este caso son de 1920x1080 frente a los 1280 x 720 pixeles de la otra base de datos. En la base de datos sintética, se empleó un modelo de cámara sin distorsión radial y tangencial. En esta base de datos, se realizó una calibración previa de la cámara empleada en la adquisición de los vídeos y refleja unos parámetros propios en cuanto a distorsión, distancia focal y coordenadas del centro óptico.

La base de datos ha sido organizada por usuario. Al igual que en el caso de la base de datos sintética, los nombres de archivo de cada vídeo describen el vídeo el número del usuario y el número de vídeo, *user_xx_video_yy.mp4*, donde *xx* se corresponde al número de usuario y *yy*, al número del vídeo, desde donde se extraen los fotogramas. Los vídeos del 1 al 3 son vídeos con movimiento en traslación *x, y* y *z*, respectivamente. El vídeo 4 se relaciona con un movimiento tipo *roll*. Los vídeos 5 y 6, movimientos puros en *yaw* y *pitch*, respectivamente. Los vídeos del 7 al 12 son movimientos arbitrarios, combinación de movimiento en cualquier componente de traslación y rotación.

La información relativa al *ground truth* 2D está contenida en archivos de texto de la forma *user_xx_video_yy_groundtruth2D.txt*. Cada uno de estos archivos contiene 300 filas, cada una correspondiente a un fotograma, y 108 columnas, que corresponde a las coordenadas X e Y de la imagen (en píxeles) de los 54 puntos de referencia: *x 1 y 1 x 2 y 2 ... x 54 y 54*. Se corresponden a los puntos GT2D, pero no están vinculados con *IntraFace*.

Los 43 puntos detectados con *IntraFace* para cada fotograma con y sin promediado están almacenados en un archivo, siguiendo el formato *user_xx_video_yy_intraface2.mat*, para los puntos promediados en 20 iteraciones y *user_xx_video_yy_intraface3.mat*, para los puntos sin promediar, donde *xx* se corresponde al número de usuario y *yy*, al número del vídeo.

La información relativa al *ground truth* de HPE está contenida en archivos de texto de la forma *user_xx_video_yy_groundtruth3D.txt* Consisten en 300 filas y 6 columnas, correspondientes a las 3 traslaciones y 3 rotaciones, ordenados como T_x, T_y, T_z, *roll*, *yaw*, *pitch*. Traslaciones en milímetros y rotaciones en grados.

model3D

54x3 double

	1	2	3
1	-58.6284	113.2681	-51.6511
2	-46.7781	104.3389	-72.5386
3	-31.3408	102.6449	-82.6304
4	-18.4297	105.1186	-86.1984
5	-15.2111	113.2669	-82.5309
6	56.5368	105.8133	-53.5215
7	48.3077	101.7742	-68.4126
8	30.5136	102.2753	-81.2486
9	17.2761	106.6395	-85.7078
10	14.1727	114.9201	-81.4436

Figura 2.3. Ejemplo del contenido del archivo *user_01_video_06_groundtruth3D.txt*.

Los archivos del modelo de la cara 3D real o GT3D se nombran como *user_xx_video_yy_model3D.txt*. Contienen 54 filas, cada una correspondiente a un punto facial en tres dimensiones, y 3 columnas, que corresponde a las coordenadas *x, y,* y *z*. Los valores de las coordenadas se expresan en milímetros.

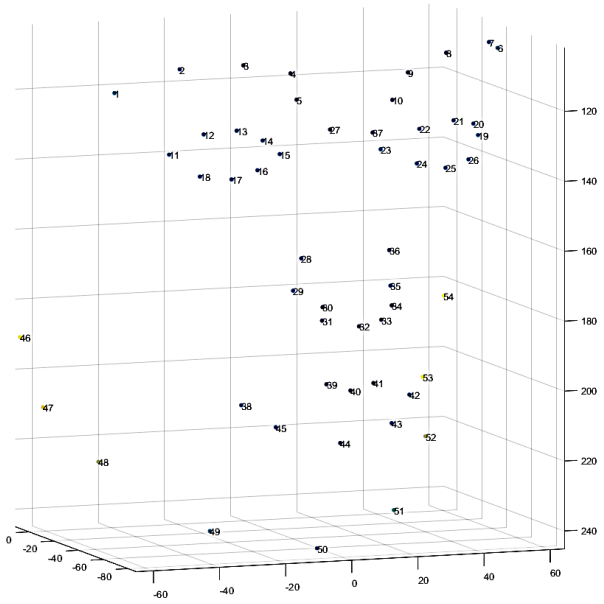


Figura 2.4. Representación espacial del modelo 3D del usuario 1.

2.2. Diseño del algoritmo de reconstrucción.

2.2.1. Funciones y algoritmos de MATLAB 2016a aplicado a SfM.

La librería de SfM de MATLAB 2016 incluye una amplia lista de funciones y aplicaciones orientadas tanto a la calibración de una única cámara o estéreo, obtención de puntos característicos de una imagen y extracción de sus descriptores, reconstrucción 3D a partir de las localizaciones de los puntos, así como obtención de las poses relativas entre dos imágenes.

En las anteriores versiones de MATLAB, SfM estaba orientado principalmente a trabajar con imágenes en estéreo, es decir, solo dos puntos de vistas. Sin embargo, MATLAB 2016 incluye funciones y objetos que nos permiten trabajar con múltiples vistas, además de incluir otras que nos permiten mejorar la reconstrucción obtenida, tal como la función de *bundleAdjustament*. Trabajar con múltiples vistas puede ser la incorporación más destacable de esta nueva versión respecto a sus predecesoras.

A continuación, se describen todo el conjunto de funciones extraídas de la documentación de MATLAB 2016a relacionadas con SfM.

Aplicaciones

Camera Calibrator	Calcular los parámetros geométricos de una sola cámara.
Stereo Camera	Estimar los parámetros geométricos de una cámara estéreo.

Funciones

bundleAdjustment	Refinar las poses de la cámara y los puntos 3D.
cameraMatrix	Matriz de proyección de la cámara.
cameraPose	Calcular la rotación y traslación relativa entre poses de la cámara.
epipolarLine	Calcular líneas epipolares de imágenes estéreo.
estimateCameraParameters	Calibrar una sola cámara o estéreo.
estimateFundamentalMatrix	Estimado de la matriz fundamental a partir de puntos correspondientes en imágenes estéreo.
extrinsics	Calcula la ubicación de una cámara calibrada.
isEpipoleInImage	Determinar si la imagen contiene puntos epipolares.
lineToBorderPoints	Puntos de intersección de líneas en la imagen y en el borde la imagen.
rotationMatrixToVector	Convertir matriz 3D de rotación en vector de rotación.
rotationVectorToMatrix	Convertir vector 3D de rotación en matriz de rotación.
triangulate	Ubicaciones en 3D de puntos coincidentes sin distorsión en las imágenes estéreo.
triangulateMultiview	Ubicaciones en 3D de puntos emparejados sin distorsión a través de múltiples imágenes.
undistortImage	Corregir una imagen por distorsión de la lente.
undistortPoints	Corregir coordenadas de los puntos por distorsión de la lente.
pcshow	Representa la nube de puntos 3D.
plotCamera	Representa una cámara en coordenadas 3D.
detectBRISKFeatures	Detectar características BRISK y devolver objeto BRISKPoints.
detectFASTFeatures	Detectar esquinas FAST y devolver objeto cornerPoints.
detectHarrisFeatures	Detectar esquinas usando el algoritmo de Harris-Stephens y devolver objeto cornerPoints.
detectMinEigenFeatures	Detectar esquinas usando el mínimo valor singular y devolver objeto cornerPoints.
detectMSERFeatures	Detectar características MSER y devolver un objeto MSERRegions.
detectSURFFeatures	Detectar características SURF y devolver un objeto SURFPoints.
extractFeatures	Extraer de descriptores de los puntos de interés.

extractHOGFeatures	Extraer el histograma de las características de gradientes orientados (HOG).
matchFeatures	Encontrar las características emparejadas.

Clases

cameraParameters	Objeto para almacenar parámetros de la cámara.
cameraCalibrationErrors	Objeto para almacenar los errores estándar de los parámetros estimados de la cámara.
pointTrack	Objeto para almacenar puntos coincidentes desde múltiples puntos de vista.
viewSet	Objeto para la gestión de datos para SfM y de odometría visual.

Objetos de sistema

vision.PointTracker	Seguimiento de los puntos en un vídeo utilizando el algoritmo Kanade-Lucas-Tomasi (KLT).

2.2.2. Funcionamiento del algoritmo de reconstrucción.

Conocido el conjunto de información que necesita MATLAB para realizar la reconstrucción a partir de cualquier vídeo, se concibe un algoritmo que permita ajustarnos a su flujo de trabajo introduciendo cualquier información relativa a los puntos característicos encontrados en cada fotograma y las posiciones relativas de la cámara en la forma que se especifica en cada caso.

Trabajar con las posiciones relativas de la cabeza que tenemos de partida a través del *ground truth* nos permite conocer si la triangulación multivista realmente funciona y podemos emplearlo para diseñar un algoritmo genérico para cualquier modelo de cabeza.

Sin embargo, el procedimiento se diseña con vistas a ser empleado con modelos de cabeza reales, donde es posible que no se disponga del *ground truth*, es decir, que no conozcamos de partida la posición de la cabeza en términos de rotación y traslación, y, por tanto, la reconstrucción nunca va a ser ideal. Por ello, el algoritmo que se presenta a continuación tiene la ambición de poder funcionar con la información mínima necesaria en la introducción al sistema, como son los puntos característicos de cada imagen y la matriz de calibración de la cámara.

El procedimiento está basado en la idea principal de reconstrucción a través de unas poses de cámara a partir de la estimación de la posición de la cabeza en cada fotograma del vídeo mediante un algoritmo denominado POSIT (*Pose from Orthographic and Scaling with ITerations*). Dicho algoritmo calcula la transformación que debe realizarse a un modelo de cabeza tridimensional para que su proyección en un plano tenga el mínimo error cuadrático respecto a los puntos característicos localizados en cada fotograma. Esto nos permitirá conocer la estimación de la posición de la cabeza o *Head Position Estimation* (HPE).

POSIT necesita un modelo de cabeza de referencia para hallar la estimación. Si a priori no se conoce el modelo tridimensional del usuario, debemos buscar un modelo que nos permita emplearlo para realizar la proyección de los puntos característicos y obtener una estimación aproximada. Nuestro algoritmo genera esta referencia a partir de la cabeza media de BFM.

Los puntos detectados en la base de datos corresponden a los puntos característicos faciales detectados por el algoritmo de *IntraFace*.

IntraFace es un software de detección y seguimiento de puntos característicos de la cara desarrollado en la Universidad de Carnegie Mellon y la Universidad de Pittsburgh, que detecta 49 puntos faciales característicos situados alrededor de las cejas, nariz y labios, que facilita enormemente el proceso de

seguimiento de esos puntos a lo largo de los fotogramas sucesivos al tratarse de un algoritmo específico para el modelo de una cabeza.

En aplicaciones prácticas, se reduce el número de puntos a 43, descartando aquellos 6 puntos situados en el interior del labio por considerarse problemáticos en nuestro procedimiento.

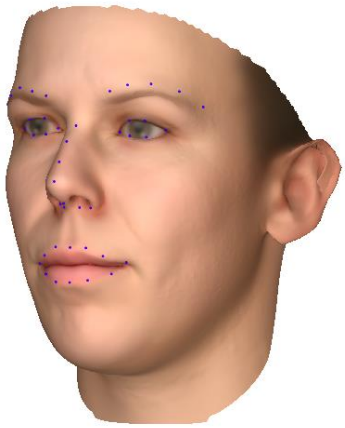


Figura 2.5. Puntos de *IntraFace* localizados en un modelo de cabeza sintética.

Si tenemos disponibles los puntos de *IntraFace* en cada fotograma para cada vídeo, estamos trabajando con un total de 43 puntos. Para realizar POSIT, se debe tener indexada de forma adecuada la cabeza media generada mediante BFM, de forma que tengamos total correspondencia de los puntos característicos de *IntraFace* vinculados a la cabeza media para obtener de este modo nuestro *ground truth*. Para hallar esta correspondencia entre los puntos de BFM e *IntraFace*, Miguel Gamallo generó con el simulador una imagen frontal de la cabeza y se detectaron los puntos característicos con *IntraFace*. A continuación, se buscó el punto más cercano de los 53490 que forman BFM para cada uno de los 43 puntos de *IntraFace*, buscando los puntos más cercanos en los modelos de BFM [12]. Miguel obtuvo todos los índices de *IntraFace* correspondientes a los modelos tridimensionales de los 10 usuarios incluyendo el de la cabeza media. El resultado de indexar los 53490 puntos puede verse en la siguiente figura para el caso de la cabeza media.

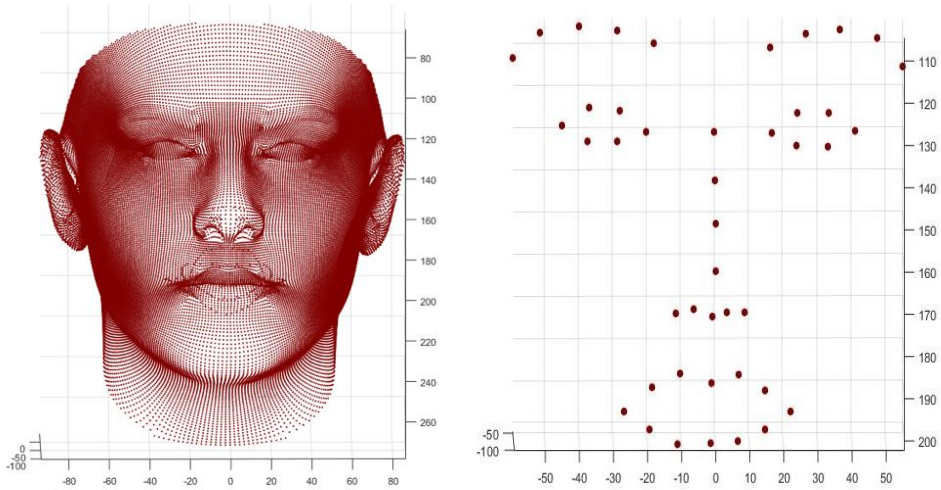


Figura 2.6. Representación tridimensional de la cabeza media de BFM. A la izquierda, cabeza media completa, a la derecha, cabeza indexada con los 43 puntos de *IntraFace*.

Pero conocer el HPE en cada fotograma no es suficiente, ya que necesitamos conocer las poses relativas de la cámara, es decir, debemos suponer un cambio en el sistema de referencia del escenario del

vídeo. Se requiere que la cámara deje de estar inmóvil y sea el modelo de cabeza el que adopta esta posición pasiva. Esto se realiza aplicando una transformación a las matrices obtenidas mediante POSIT para llegar a este escenario relativista. En los vídeos, la cámara siempre está inmóvil.

La transformación se realiza de la siguiente manera:

1. Se obtiene un vector de traslación, *T*, y una matriz de rotación, *R*, procedente de aplicar POSIT para cada fotograma, transformación que hay que aplicar a la cabeza tener una determinada HPE.

2. Se construye una matriz compuesta por *T* y *R* que denominamos *M*.
$M = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$

3. M_1 corresponde a la matriz del primer fotograma. Las posiciones relativas de la cámara se hacen respecto a este primer fotograma. Para cualquier fotograma *n*, se realiza la siguiente operación.
$M_{relativo,n} = M_1 * (M_n)^{-1}$

4. Se extraen los vectores *R* y *T* de la matriz obtenida para cada fotograma.

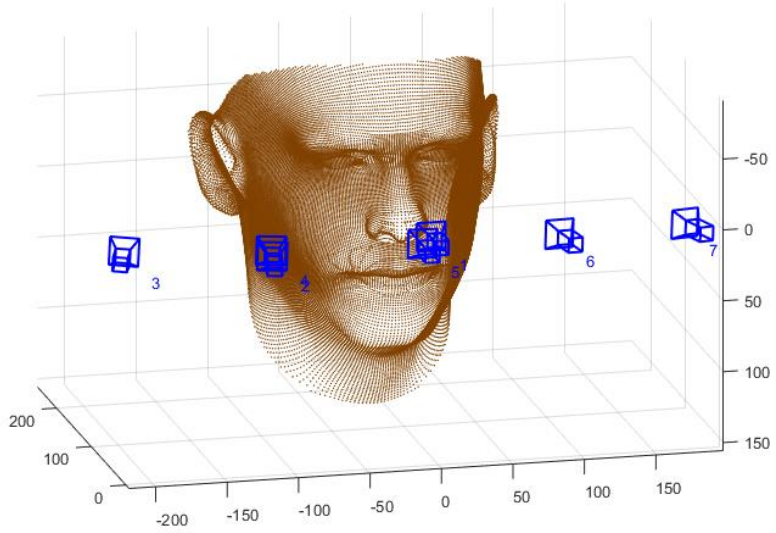


Figura 2.7. Cambio a poses relativas de cámara en el primer usuario con movimiento de componente *yaw* y diferencia entre fotogramas de 16º en esta componente.

La estimación de la posición de la cabeza nos genera un error en la reconstrucción si lo comparamos con el modelo real, por ello es necesario construir una rutina para refinar el modelo obtenido, con el fin de minimizar dicho error:

1. Generar el modelo medio de BFM. Se escoge este modelo porque está parametrizado a partir de 200 caras con mucha resolución de puntos. Está completamente definido e indexado. Sus características son promedio, lo que lo hace más independiente respecto al modelo de cabeza de entrada que vayamos a reconstruir, lo que genera un error de estimación de posición de cabeza similar, en cualquier caso.

2. Buscar los índices que emplea *IntraFace* para la detección de puntos faciales correspondientes en el modelo medio en todas las imágenes. Pueden estar precalculados o se pueden hallar con sus respectivas funciones.

3. Aplicar POSIT a los índices de *IntraFace* para cada fotograma. Nos permite calcular la transformación que se ha aplicado al modelo medio centrado en el origen para desplazarlo a la

posición relativa de nuestra cabeza en cada fotograma. POSIT nos devuelve las matrices de transformación que se ha aplicado a nuestra nube de puntos respecto al modelo centrado.

4. Generar un conjunto de vistas que incluya para cada vista las poses relativas de la cámara, puntos de *IntraFace* y las conexiones o correspondencias de puntos entre fotogramas.

5. El modelo obtenido anteriormente se refina mediante un proceso de optimización denominado *Bundle Adjustment* para minimizar el error de proyección en cada fotograma. A su vez, se refinan las matrices de transformación obtenidas mediante POSIT.

6. El último paso es registrar la nube de puntos optimizada en cada iteración mediante *Procrustes*. Permite obtener una traslación y rotación que hay que aplicar a la cabeza original para alinearla con la segunda cabeza, de esta forma, se obtiene la media de error entre ambas, *Mean Square Error* (MSE) y se compara con un cierto umbral: un límite para considerar nuestro modelo lo suficientemente bueno para dar por finalizado el proceso iterativo. Si no se alcanza el umbral, se prosigue refinando el modelo mediante el procedimiento anterior desde el punto 4 para minimizar el MSE. El proceso iterativo continúa mientras el MSE sea menor respecto a la anterior iteración o no se haya alcanzado un número máximo de iteraciones.

Para complementar la visualización y entendimiento de lo descrito anteriormente, se transcribe a pseudocódigo con el fin de mostrar el correcto flujo del algoritmo:

```
Cargar la cabeza media
frames, parámetros de la cámara, model3D ideal <= userReader(número de usuario, máximo ángulo de rotación entre
frames, tipo de puntos de IntraFace)

iteración = 1
converged = false

while converged == false
     for i=1: número de frames
               if iteración==1
                         orientación, localización <= HPE (puntos IF del frame i, puntos IF de mean head,...
                         parámetros de la cámara)

               else
                         orientación, localización <= HPE (puntos IF del frame i, modelo reconstruido anterior,
                         parámetros de la cámara)
                         end
     end

     camera_poses <= relativePoses_HPE (HPE de todos los frames , parámetros de la cámara)

     crear set de vistas
     añadir vista (1, puntos IF del frame 1, orientación relativa 1, localización relativa 1)

     for i = 2: número de frames
               añadir_vista (i, puntos IF del frame i, orientación relativa i, localización relativa i)
     end

     modelo reconstruido <= triangulateMultiview (vistas, camera_poses, parámetros de la cámara)

     modelo refinado <= bundleAdjustment (modelo reconstruido, vistas, camera_poses, parámetros de la cámara)

     if iteración==1
                         MSE_anterior <= procrustes (modelo 3D ideal, modelo refinado)
                         Modelo reconstruido original <= modelo refinado
     else
                         MSE_actual <= procrustes (model3D ideal, modelo refinado)

     if MSE_actual/MSE_anterior>1 e iteración>=2
                                   converged = true
                         else
                                   modelo reconstruido anterior <= modelo refinado
                                   MSE_anterior <= MSE_actual
                         end
     end
     sumar 1 a la variable de iteración.
end
```

Figura 2.8. Pseudocódigo del algoritmo.

Procrustes permite hacer el cálculo escalando la segunda nube de puntos al tamaño de la primera nube. Puede resultar interesante en el proceso porque en SfM no disponemos de información de escala en los vectores de traslación o rotación. En este caso, se permite el escalado y se compara la nube de puntos

ideal de un usuario determinada con la nube de puntos procedente de la reconstrucción mediante el procedimiento.

A continuación, se muestra el diagrama de flujo seguido a lo largo del algoritmo a modo de resumen del proceso completo:

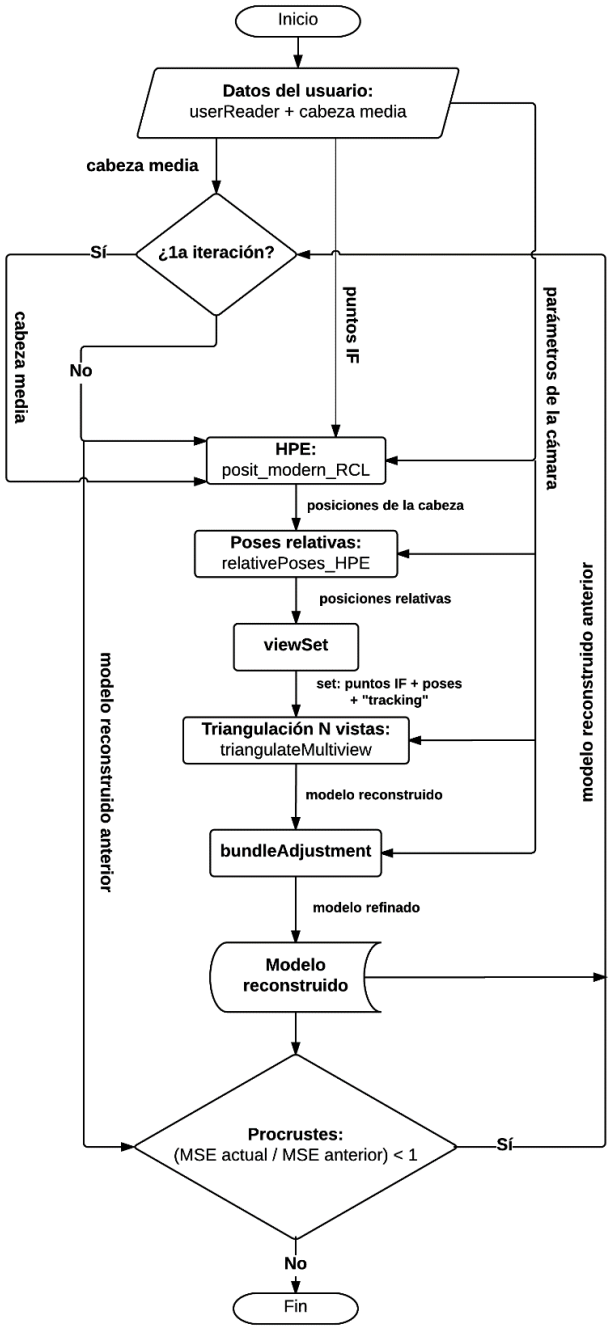


Figura 2.9. Diagrama de flujo del método implementado.

Deseamos obtener una estimación bastante aproximada del funcionamiento del algoritmo. Se han realizado las primeras pruebas utilizando imágenes entrelazadas con movimiento puro en componente *yaw* en los fotogramas impares y con movimiento puro en componente *pitch* en los fotogramas pares. La diferencia entre fotogramas en cada componente es de 1º, suficiente para alcanzar un compromiso entre resolución en la reconstrucción y eficiencia computacional.

2.3. Valoración de los resultados obtenidos. Base de datos sintética.

2.3.1. MSE, error cuadrático medio.

Aplicando primeramente el procedimiento previamente expuesto a la base de datos sintética, se obtiene una reconstrucción para cada usuario a la salida de la primera iteración del algoritmo. Posteriormente, dicha reconstrucción se emplea para obtener de nuevo unas poses de cámara refinadas respecto a las poses obtenidas mediante el modelo medio, con el fin de mejorar la reconstrucción. De tales reconstrucciones, antes y después de iterar, se plantean dos alternativas que nos permiten valorar los resultados obtenidos.

El primer planteamiento es estimar el error que existe en la ubicación de los puntos de la reconstrucción respecto al modelo tridimensional que se ha generado por medio de BFM. Es el mismo modelo que se emplea para generar los vídeos y lo denominamos *ground truth* 3D (GT3D).

El error se calcula como la distancia que existe entre el punto p'_i del modelo reconstruido con el punto p_i del modelo GT3D cuando se ha realizado la alineación de ambos modelos por medio del algoritmo de *Procrustes*, un conjunto de herramientas matemáticas de mínimos cuadrados para estimar directamente y realizar la transformación de rotación y traslación y, opcionalmente escalado, entre los puntos de coordenadas de un modelo matricial hasta su ajuste máximo con otro modelo. El error cuadrático medio o *Mean Square Error* (por sus siglas en inglés, MSE) es la distancia promedio de los *n* puntos que conforma cada modelo, en este caso, la reconstrucción tiene los 43 puntos correspondientes de *IntraFace*, por tanto, $n = 43$. Es importante matizar que debe existir una correspondencia de puntos entre los dos modelos a la hora de definirlos de manera matricial; es decir, tienen que ser correlativos. Esta condición debe cumplirse también para el algoritmo de HPE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p'_i - p_i)^2$$

Ecuación 2.4. MSE. Cálculo del error cuadrático medio.

Para realizar la reconstrucción, se pueden tomar tres tipos de puntos: puntos de *IntraFace* ideales o GT2D, promediados y sin promediar. El GT2D son aquellos que se extraen del modelo GT3D en la posición que toma en cada fotograma indexando los mismos puntos de *IntraFace*. Al ser ideales, sólo tienen validez cuantitativa, son puntos que no se pueden extraer de cualquier vídeo porque hace falta el modelo real. Esta reconstrucción valora, no tanto el funcionamiento del algoritmo, sino el error que se produce en la triangulación, los parámetros de la cámara o funciones adicionales, como el *Bundle Adjustment*. Las reconstrucciones que tienen mayor relevancia son aquellas que se realizan con puntos extraídos a partir de los vídeos como es el caso del algoritmo de *IntraFace*. De estos puntos existen dos tipos: promediados y sin promediar.

Se hace esta distinción entre puntos promediados y sin promediar a raíz de la observación de una componente de ruido inherente en el proceso de obtención de los puntos de *IntraFace*, lo que provoca que tengamos discrepancias en un mismo fotograma si se aplica en múltiples ejecuciones, como se comenta en la memoria de Rubén Segura.

La siguiente figura muestra la naturaleza de esta variabilidad para la posición frontal del primer usuario de la base de datos sintética, donde realizando 100 ejecuciones del algoritmo de *IntraFace*, se genera una deriva en la localización de los 43 puntos, con menor o mayor variación dependiendo del punto observado.

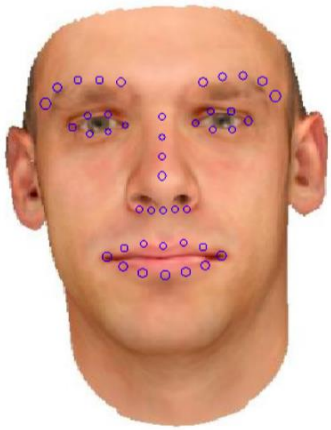


Figura 2.10. Variabilidad al ejecutar *IntraFace* 100 veces a un mismo fotograma para el usuario 1 [1].

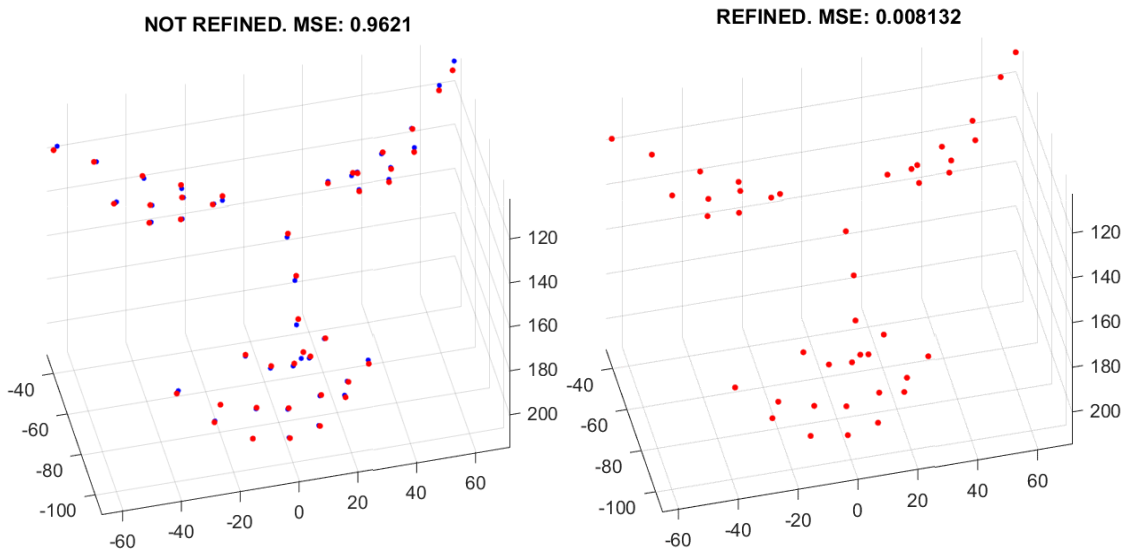
Las diferencias entre unos puntos y otros no son muy grandes, aunque se puede apreciar que la variabilidad es mayor en los puntos de las cejas y la boca.

Los puntos promediados es el resultado de promediar la localización de cada punto tras 20 ejecuciones de *IntraFace* para cada fotograma. Los puntos sin promediar tienen esta componente de ruido, al ser ejecutados solo una vez en cada fotograma.

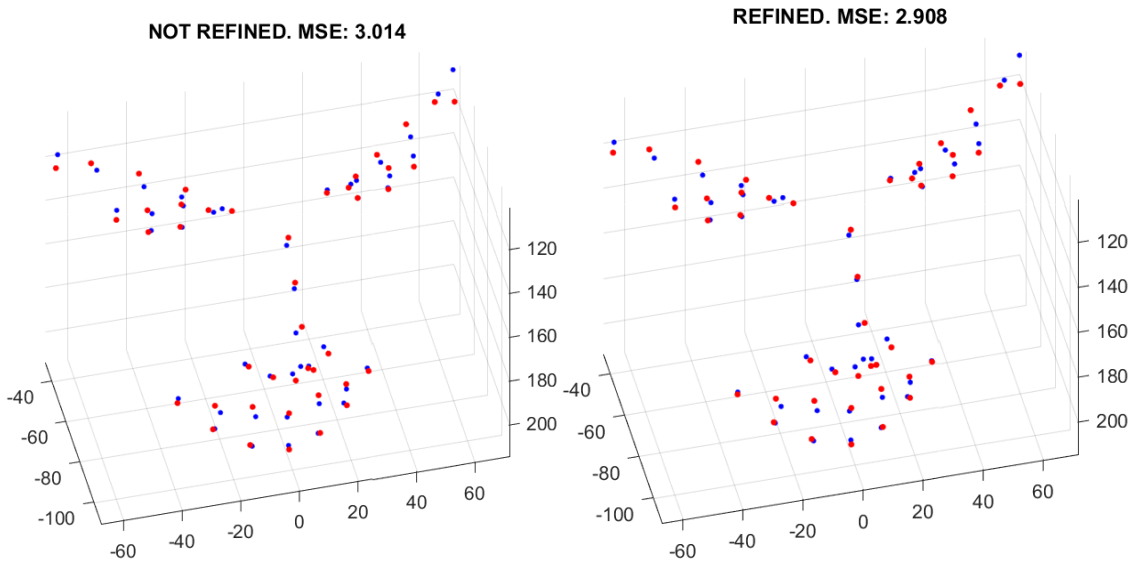
A continuación, se muestran los resultados obtenidos en términos de MSE para todos los usuarios de la base de datos sintética. Para cada usuario, se realiza un análisis de *Procrustes* del modelo reconstruido en la primera iteración respecto al GT3D y otro análisis de *Procrustes* para el modelo reconstruido tras el proceso de iterado donde se van refinando las poses y se comparan ambos MSE para valorar la eficacia del proceso. Las siguientes figuras representan las reconstrucciones obtenidas (puntos en color rojo) alineadas y escaladas mediante *Procrustes*, frente al GT3D (color azul), junto con el MSE resultante, tanto para puntos ideales o GT2D como puntos de *IntraFace* promediados.

Usuario 01:

Puntos IF GT2D:

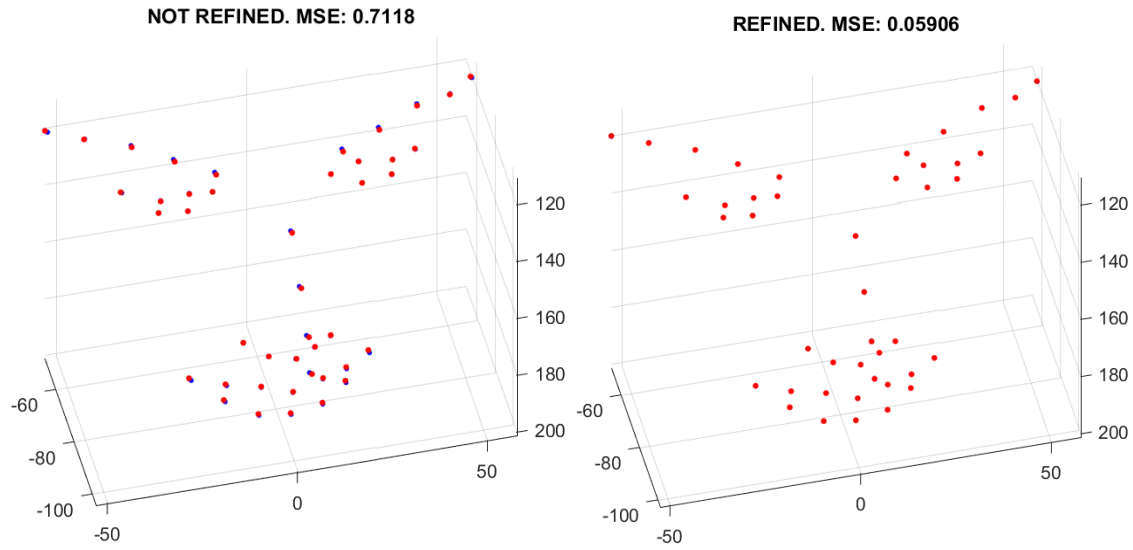


Puntos IF promediados:

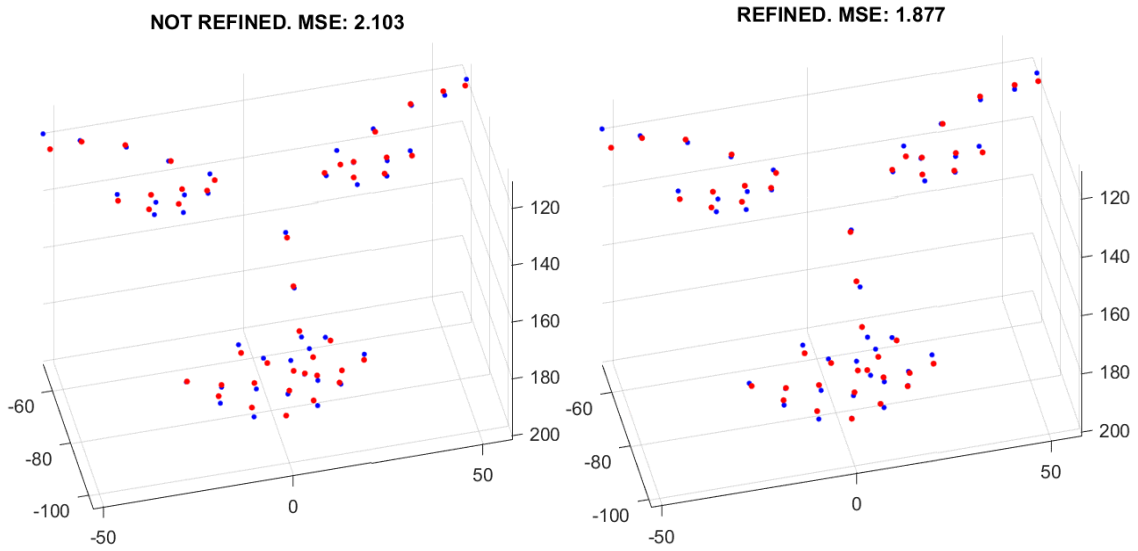


Usuario 02:

Puntos IF GT2D:

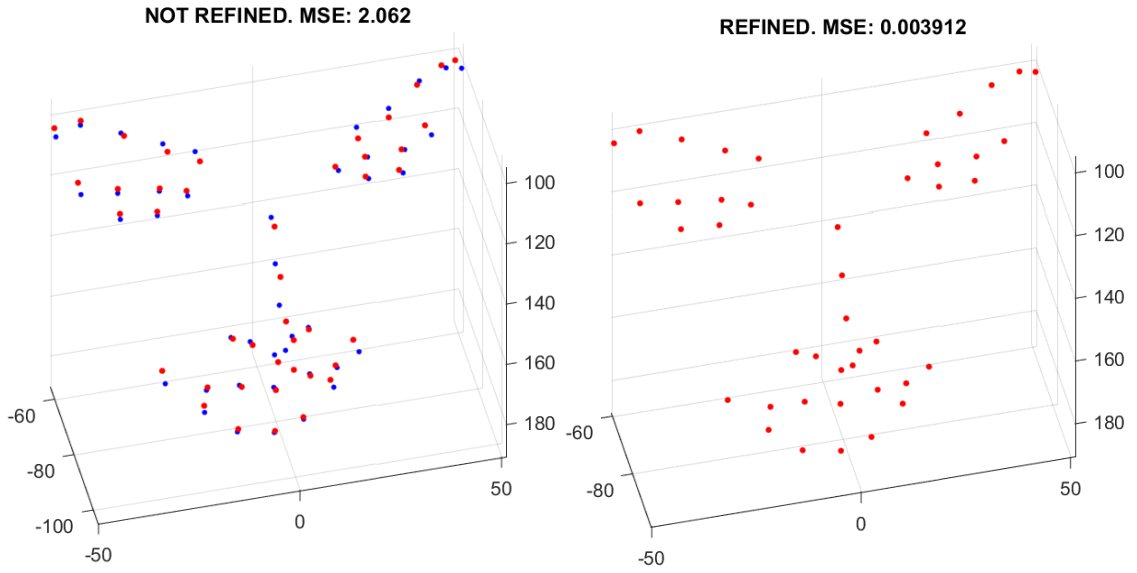


Puntos IF promediados:

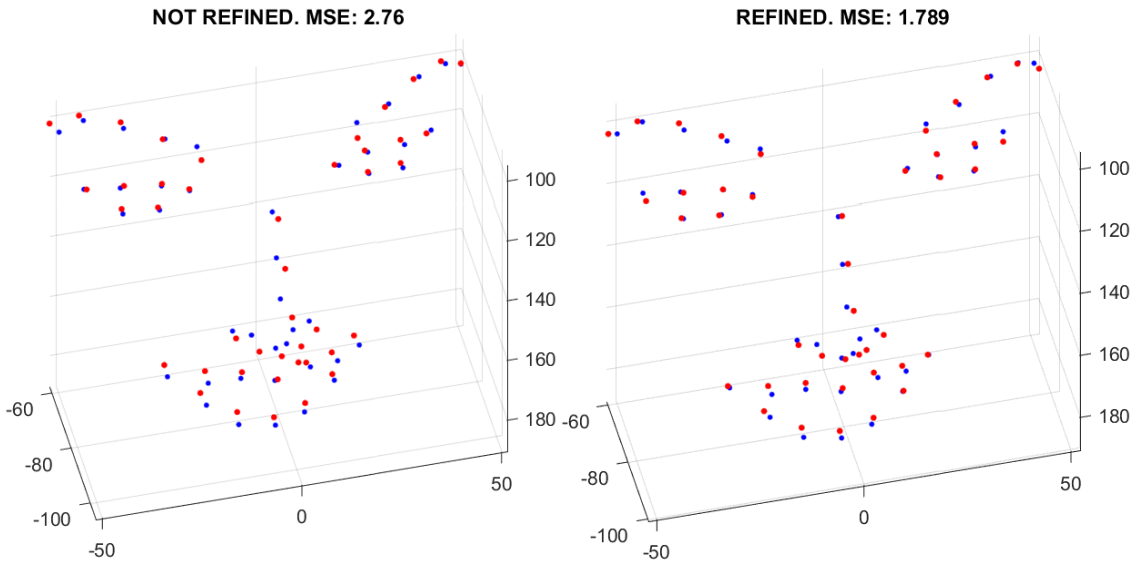


Usuario 03:

Puntos IF GT2D:

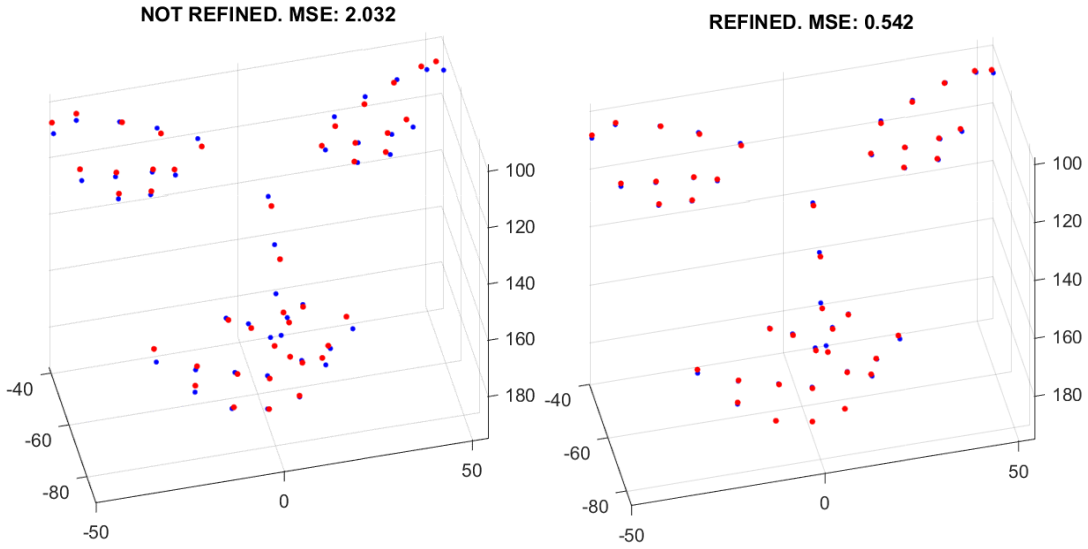


Puntos IF promediados:

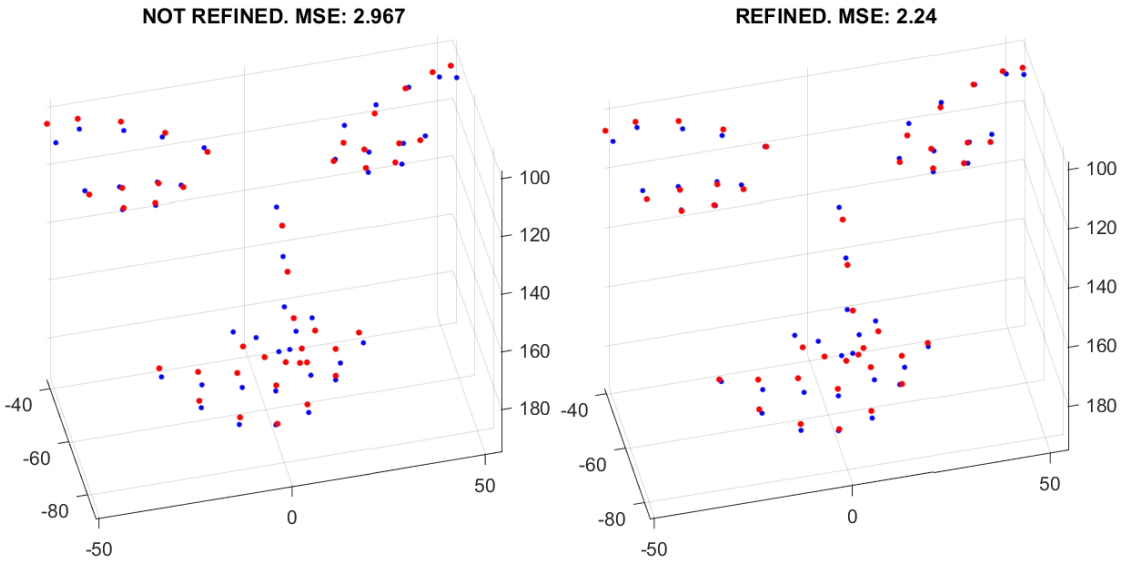


Usuario 04:

Puntos IF GT2D:

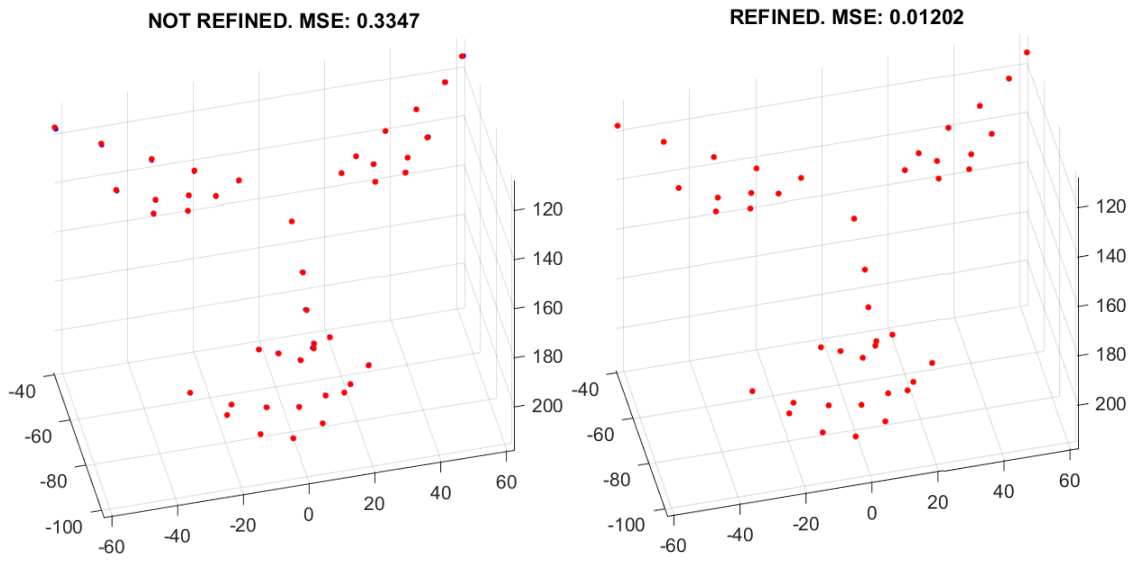


Puntos IF promediados:

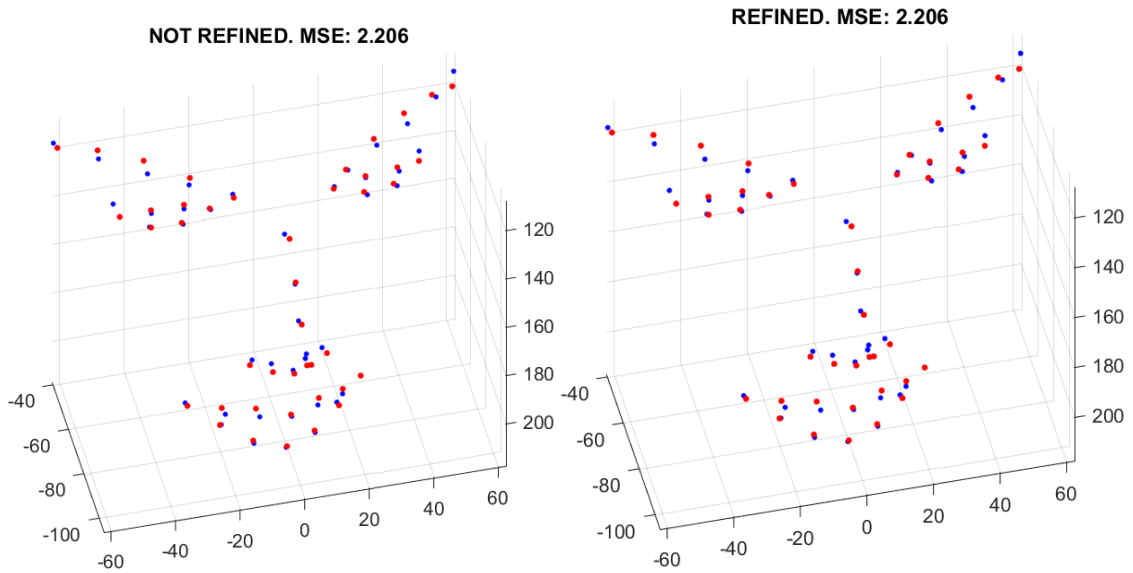


Usuario 05:

Puntos IF GT2D:

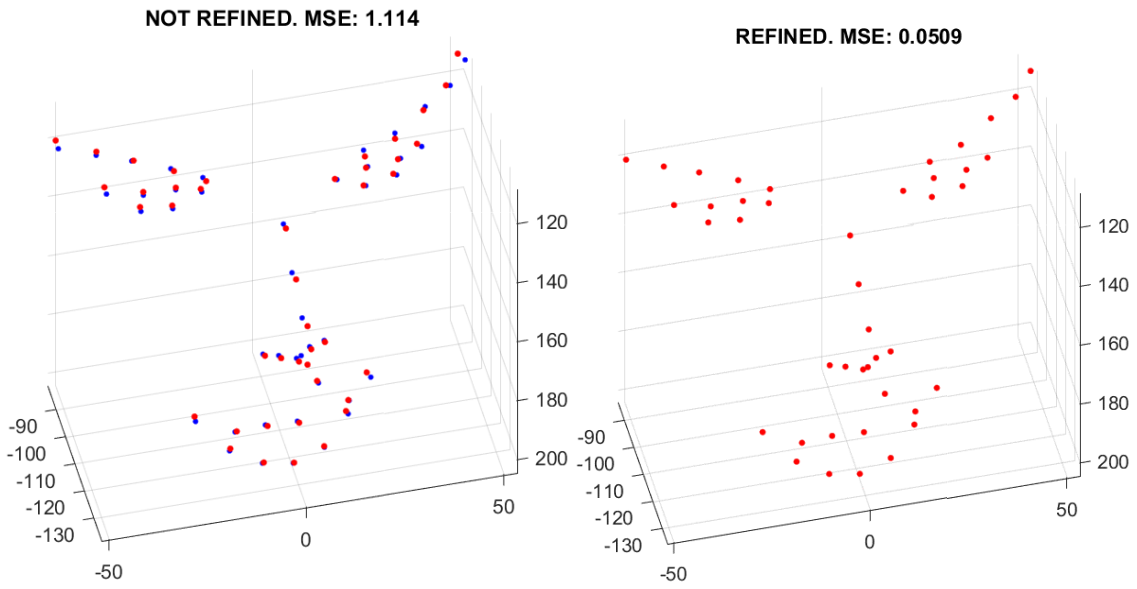


Puntos IF promediados:

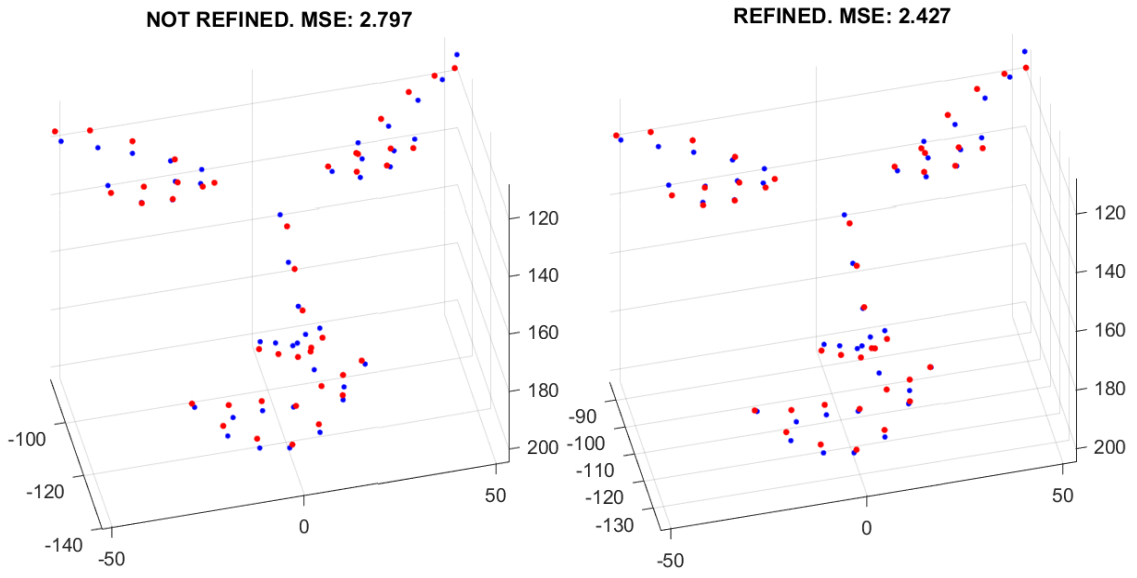


Usuario 06:

Puntos IF GT2D:

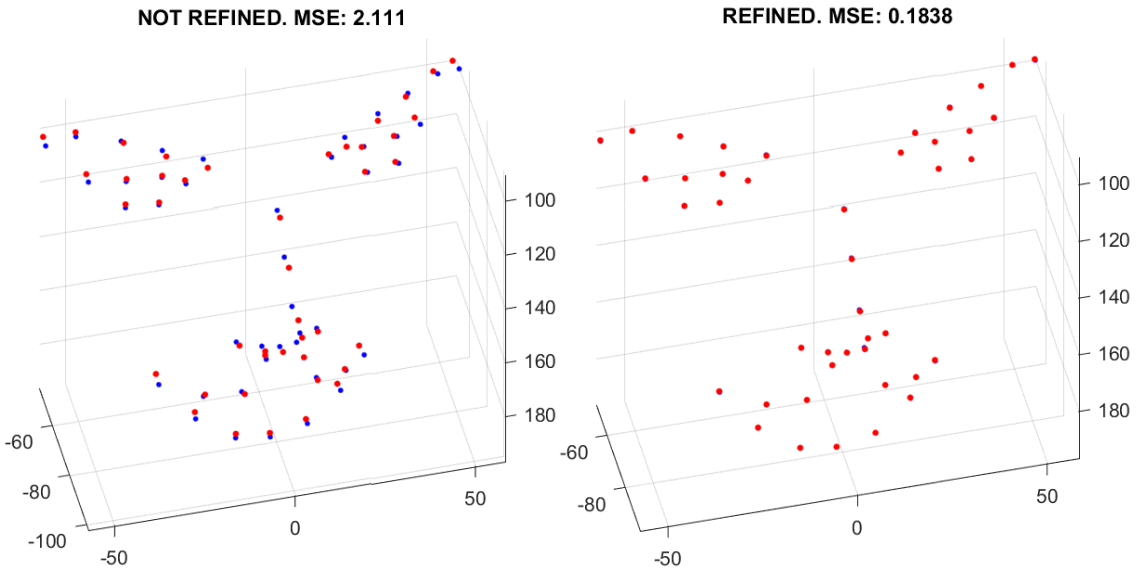


Puntos IF promediados:

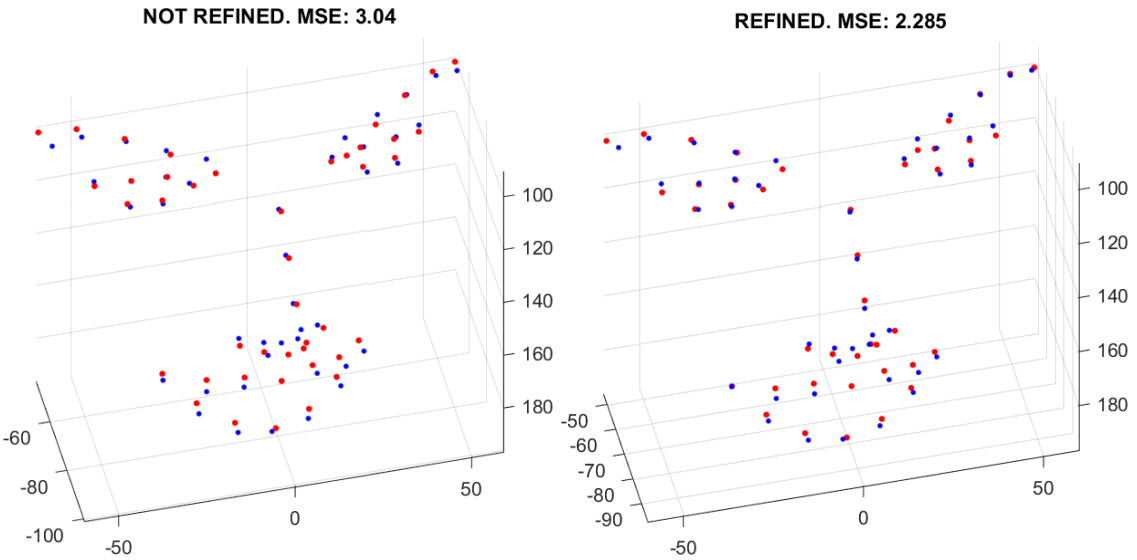


Usuario 07:

Puntos IF GT2D:

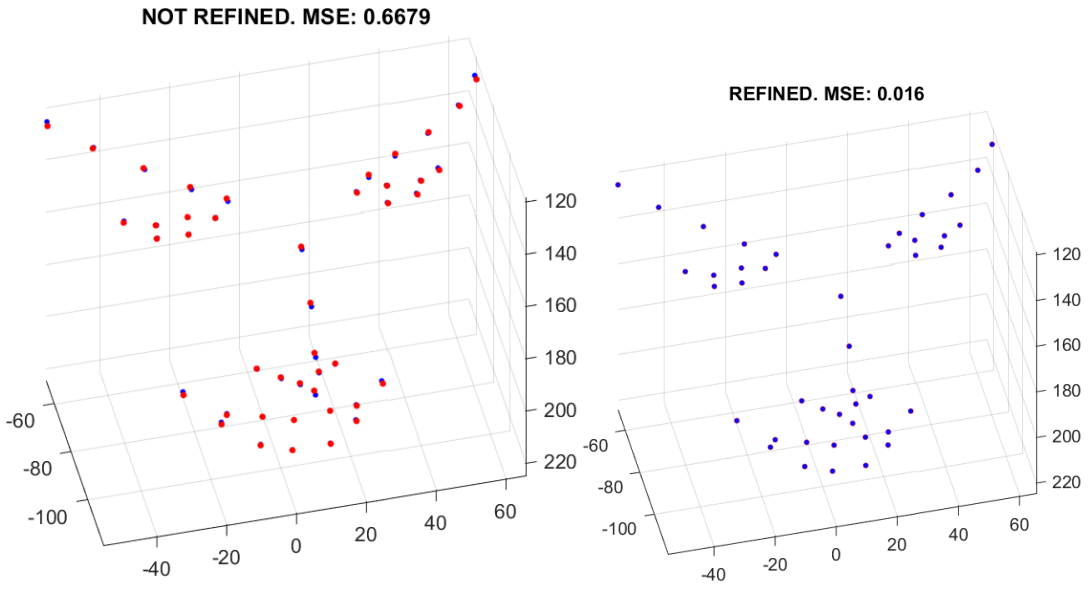


Puntos IF promediados:

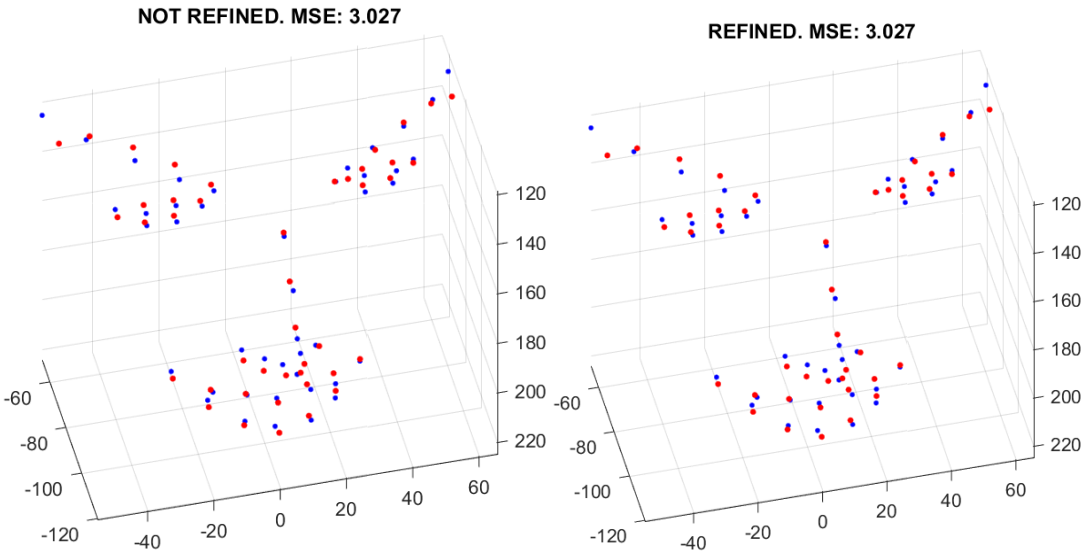


Usuario 08:

Puntos IF GT2D:

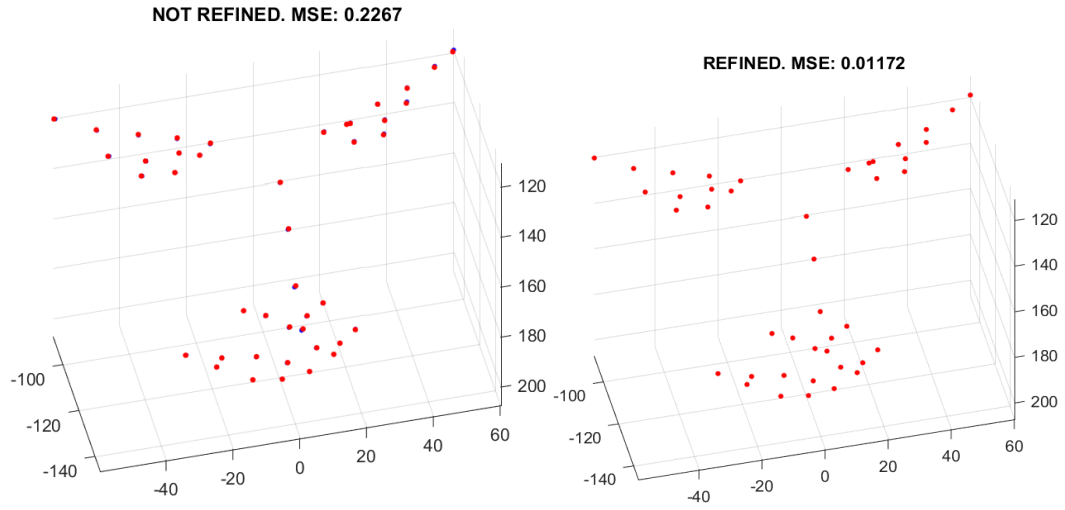


Puntos IF promediados:

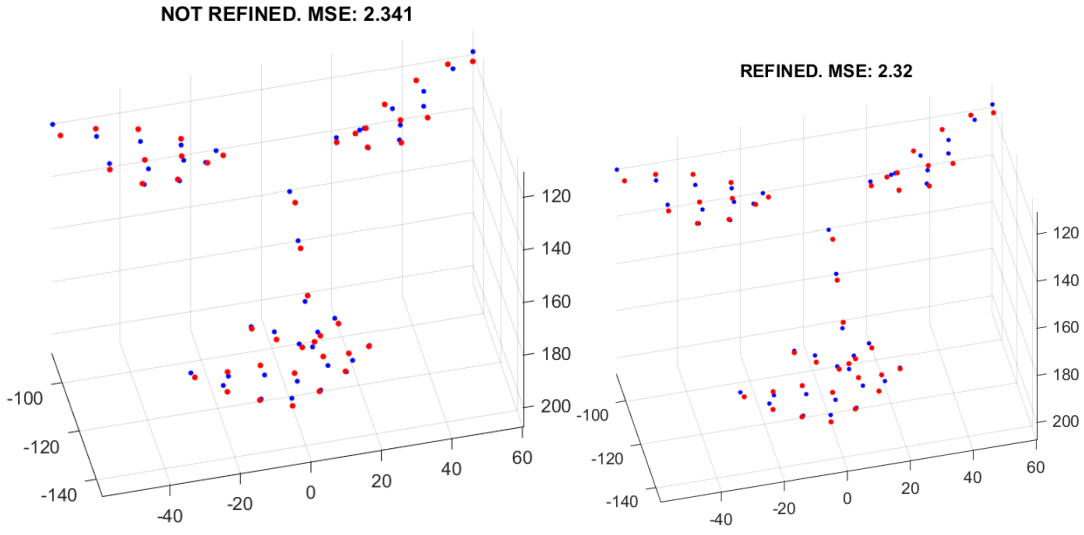


Usuario 09:

Puntos IF GT2D:

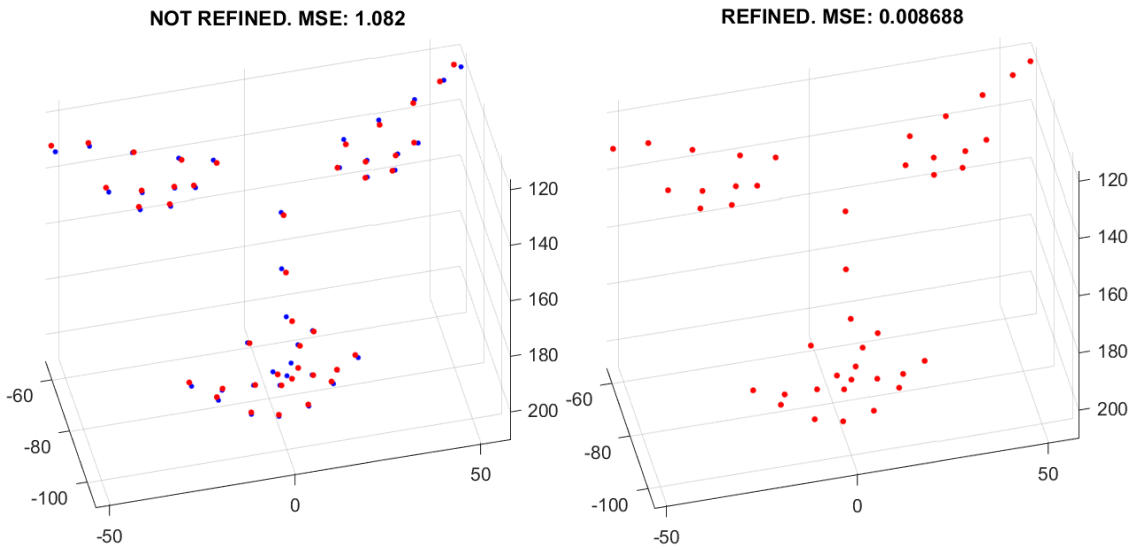


Puntos IF promediados:

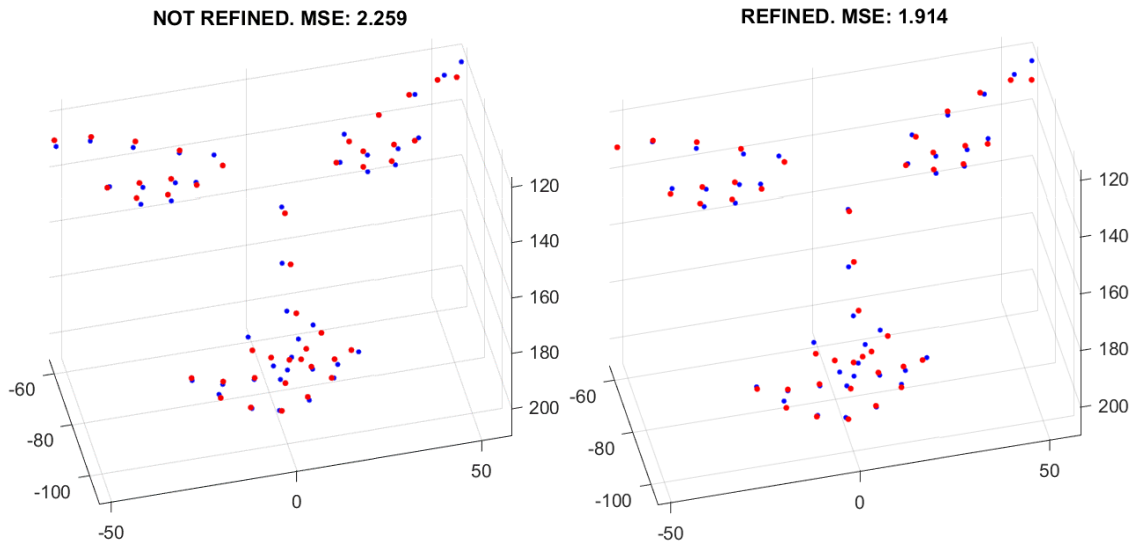


Usuario 10:

Puntos IF GT2D:



Puntos IF promediados:



Puntos de *IntraFace* ideales o GT2D.

A raíz de las figuras anteriores, podemos estimar el margen de mejora en cada caso gracias al proceso iterativo. Cabe comentar que como partimos de modelos reconstruidos escalados al tamaño del modelo de cabeza ideal, el MSE de entrada entre el modelo reconstruido y el modelo ideal es muy pequeño, siendo su unidad métrica, en milímetros.

Se muestran los resultados obtenidos en este escenario, donde se indica en la siguiente tabla cada usuario de la base sintética, las iteraciones (columna **IT.**) que han sido necesarias para llegar a la reconstrucción final, el MSE en milímetros que hay entre la cabeza media y el modelo ideal (columna **MH – GT3D**), el que existe entre el modelo ideal y el reconstruido antes de iterar (columna **1st IT. – GT3D**), el MSE entre el modelo después del proceso de iterar con el modelo ideal (columna **Nth IT. – GT3D**), la diferencia entre ambas reconstrucciones (columna **DIFF.**) y la distancia mínima (columna **MIN**) y máxima (columna **MAX**) entre dos puntos equivalentes entre el modelo refinado y el modelo 3D ideal de cada usuario.

USUARIO	IT.	MH - GT3D	1st IT. - GT3D	Nth IT. - GT3D	DIFF.	MIN	MAX
1	47	3,60	0,96	0,01	0,95	0,00	0,02
2	150	3,37	0,71	0,06	0,65	0,01	0,18
3	103	4,16	2,06	0,00	2,06	0,00	0,01
4	150	3,56	2,03	0,54	1,49	0,15	1,86
5	109	2,87	0,33	0,01	0,32	0,00	0,04
6	150	3,56	1,11	0,05	1,06	0,01	0,14
7	150	3,51	2,11	0,18	1,93	0,05	0,55
8	26	3,26	0,67	0,02	0,65	0,01	0,05
9	30	3,17	0,23	0,01	0,21	0,00	0,03
10	104	3,31	1,08	0,01	1,07	0,00	0,02
MEDIA	**101,90**	**3,44**	**1,13**	**0,09**	**1,04**	**0,02**	**0,29**

Tabla 2.1. Tabla de resultados para verificar la eficacia del método. Puntos característicos ideales.

De los resultados podemos ver dos cosas relevantes. Primeramente, si el MSE antes de iniciar el proceso de iterar para refinar el modelo es inferior a 1 milímetro, como ocurre en algunos casos, la mejora que se obtendría no sería muy apreciable si seguimos refinando el modelo, pero vemos que casi todos los modelos después de iterar tienen una aproximación prácticamente exacta respecto al modelo ideal.

En segundo lugar, se ha establecido un límite de 150 iteraciones, lo cual no significa que hayan convergido todos los modelos, indicándonos que el proceso podría prolongarse para seguir mejorando en continuas iteraciones, pero el cambio que se obtiene entre futuras iteraciones es tan pequeño, así como la diferencia ínfima con el modelo real, aproximadamente nula, que nos indica que no tiene mucho sentido seguir refinando el modelo.

El MSE se sitúa en media en 1,13 antes de iterar y 0,09 después de iterar. Podemos llegar a la asunción que el procedimiento es correcto porque la mejora que se obtiene refinando el modelo mejorando las poses de la cámara es, en media, 12 veces mejor que el modelo obtenido en la reconstrucción obtenida en la primera iteración haciendo la estimación de las poses con la cabeza media. El MSE en la primera iteración está ligado al error producido en el cálculo de las poses mediante POSIT usando la cabeza media, porque descartamos esta fuente de error en los puntos empleados en la triangulación. Se corresponden al GT2D y no introducen variabilidad.

Como se ha comentado en el apartado anterior, este escenario nos sirve para evaluar la eficacia de las funciones que intervienen en el algoritmo diseñado, aunque no son resultados que nos indican del todo el grado de bondad de nuestra reconstrucción, ya que son puntos ideales extraídos directamente del

modelo de cabeza ideal indexando los puntos característicos de *IntraFace* vinculados a esa cabeza, por tanto, no es un escenario real.

Puntos de *IntraFace* promediados y sin promediar.

Previamente, es importante matizar que, si aplicamos el algoritmo de detección de puntos de *IntraFace* a una cabeza generada con BFM, observamos que hay una buena correspondencia de puntos entre el GT2D y esta detección, si esta tiene una posición completamente frontal. Pero al desplazar y rotar la cabeza, se puede comprobar que los puntos detectados por *IntraFace* no coindicen exactamente con los puntos 2D proyectados en el simulador, aquellos puntos ideales que conforman el GT2D al indexar la correspondencia de puntos de *IntraFace* en el modelo ideal para cada fotograma, especialmente cuando la rotación de la cabeza es mayor. A continuación, se muestra un ejemplo donde se ve claramente como el error de *IntraFace* depende en gran medida de la rotación de la cabeza. En este caso, los usuarios realizan un movimiento en *yaw* y el error aumenta de forma proporcional a este ángulo. El error se calcula como la distancia de los puntos detectados respecto a los puntos del *ground truth*.

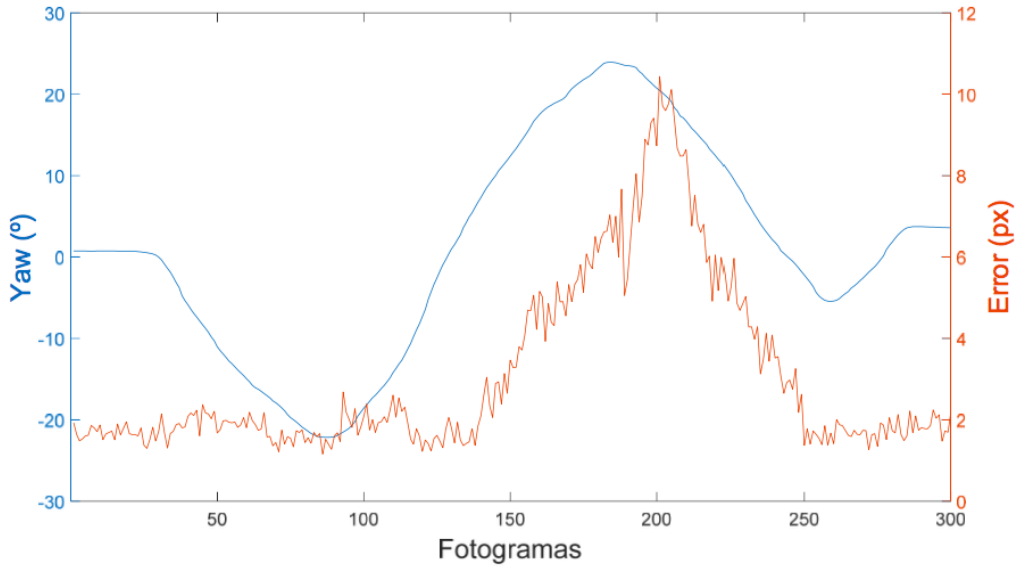


Figura 2.11. Relación entre el error de *IntraFace* y la rotación de la cabeza en *yaw* en el usuario 1 [1].

Observamos que el error aumenta conforme el ángulo de rotación es mayor.

Con el fin de poder justificar el error que podríamos llegar a tener en el límite inferior, se muestra a continuación la media del error de los puntos de *IntraFace* promediados detectados en cada fotograma respecto a sus respectivos puntos ideales o *ground truth*, para todos los usuarios.

USUARIO	DISTANCIA MEDIA (px)
1	2,23
2	1,71
3	2,22
4	1,61
5	2,26
6	2,21
7	2,12
8	1,95
9	2,17
10	2,24
MEDIA	**2,07**

Tabla 2.2. Error medio de *IntraFace* promediado los vídeos con movimiento en *yaw*.

A continuación, se muestran los resultados realizados para puntos de *IntraFace* obtenidos mediante su algoritmo de detección, aplicados directamente a los fotogramas de los vídeos, tanto promediados como ejecutados una única vez y por ello, obtenidos con una componente de ruido debido a la variabilidad del algoritmo de detección.

USUARIO	IT.	MH - GT3D	1st IT. - GT3D	Nth IT. - GT3D	DIFF.	MIN	MAX
1	119	3,60	3,01	2,91	0,11	0,36	12,15
2	21	3,37	2,10	1,88	0,23	0,29	7,01
3	150	4,16	2,76	1,79	0,97	0,20	4,32
4	150	3,56	2,97	2,24	0,73	0,32	5,76
5	2	2,87	2,21	2,21	0,00	0,31	7,08
6	141	3,56	2,80	2,43	0,37	0,35	6,50
7	150	3,51	3,04	2,28	0,76	0,36	10,61
8	2	3,26	3,03	3,03	0,00	0,30	13,32
9	4	3,17	2,34	2,32	0,02	0,36	5,59
10	14	3,31	2,26	1,91	0,35	0,39	7,25
MEDIA	**75**	**3,44**	**2,65**	**2,30**	**0,35**	**0,32**	**7,96**

Tabla 2.3. Resultados para los 10 usuarios empleando los puntos característicos de *IntraFace* promediados.

El MSE se sitúa en media en 2,65 antes de iterar y 2,30 después de iterar para el escenario con los puntos de *IntraFace* promediados, lo que supone una mejora del 11,5%.

El error que tenemos en la reconstrucción se sitúa en torno a 2 milímetros en la primera iteración antes de comenzar el proceso de refinamiento. Tras el refinado, la diferencia no es tan acusada con el modelo anterior, como ocurría en con los puntos característicos ideales, se consigue mejorar en media 0.23 milímetros, aunque en algunos usuarios como el 5 y el 8, en el proceso iterativo no se obtienen mejora y nos quedamos con el modelo obtenido antes de iterar.

Vemos que, respecto al escenario ideal, el MSE se incrementa, como se deduce lógicamente a partir de la gráfica anterior, al emplear fotogramas que se rigen por el patrón de movimiento y, por tanto, el error está relacionado tanto por la deriva introducida por el algoritmo de detección, como por el error generado en el cálculo de las poses de la cámara mediante POSIT a través de la cabeza media. Podemos deducir también que el proceso de refinamiento del modelo reconstruido tiene mayor efectividad si las poses iniciales, tras la primera iteración, son cercanas a las ideales, como se infiere de la tabla de resultados empleando puntos GT2D. En estos casos, el algoritmo admite más iteraciones, y, por tanto, se consigue reducir en mayor medida el error.

USUARIO	IT.	MH - GT3D	1st IT. - GT3D	Nth IT. - GT3D	DIFF.	MIN	MAX
1	98	3,60	3,03	2,92	0,10	0,41	12,16
2	21	3,37	2,09	1,86	0,23	0,19	7,05
3	150	4,16	2,79	1,81	0,98	0,16	4,51
4	150	3,56	3,04	2,17	0,87	0,28	5,62
5	2	2,87	2,20	2,20	0,00	0,32	7,04
6	150	3,56	2,77	2,37	0,40	0,35	6,48
7	150	3,51	3,03	2,26	0,78	0,36	10,48
8	2	3,26	3,06	3,06	0,00	0,33	13,56
9	5	3,17	2,37	2,35	0,02	0,25	5,44
10	14	3,31	2,25	1,90	0,35	0,39	7,18
MEDIA	**74,20**	**3,44**	**2,66**	**2,29**	**0,37**	**0,30**	**7,95**

Tabla 2.4. Resultados para los 10 usuarios empleando los puntos característicos de *IntraFace* sin promediar.

El MSE obtenido en este caso en media es muy similar a los obtenidos sin promediar: 2,66 antes de iterar y 2,29 después de iterar. Nos hace llegar a la conclusión que no afecta el hecho de que los puntos de *IntraFace* estén promediados para este método, porque en la práctica los resultados son idénticos y posiblemente no necesitaríamos tantas iteraciones, pues la mejora no es muy grande. Todo depende del rigor que se necesite en la reconstrucción.

2.3.2. Error en HPE.

Otra forma de evaluar la precisión del método implementado en cada apartado, consiste en comparar las poses obtenidas mediante el algoritmo de estimación frente a este *ground truth*. Esta diferencia se constituye como el error en HPE (Head Pose Estimation).

Los errores que se muestran en las tablas más adelante son el error medio para todos los fotogramas de la base de datos en traslación y rotación (*roll*, *yaw* y *pitch*). Los errores se calculan de tres formas distintas: el error absoluto en el cual se calcula la diferencia entre la estimación y el *ground truth*, el error con la estimación modificada para que sea cero en el primer fotograma de cada vídeo y el error con la estimación modificada para que esté alineada con el *ground truth* en el primer fotograma.

El motivo de modificar la estimación del error en rotación para que sea nula en rotación en el primer fotograma se basa en la consideración previa de que todos los vídeos parten de una posición neutral, siendo los tres ángulos de rotación cercanos a cero. De esta forma se minimiza la componente de error debido al desajuste entre el modelo real y el modelo reconstruido o el modelo medio. De todos modos, con esta modificación no se elimina directamente esta componente, ya que la rotación no es exactamente nula en el primer fotograma, por lo que se calcula el tercer error donde la estimación se modifica para que en el primer fotograma el error sea nulo, pasando de tener un error absoluto a un error diferencial. En el segundo tipo de error no tiene sentido calcular el error de traslación, ya que no se puede hacer una corrección genérica a la estimación de la traslación en el primer fotograma debido a que en cada vídeo la posición inicial es diferente.

Es importante recalcar que el modelo reconstruido será, en escala, parecido al modelo empleado en la estimación de la pose de la cabeza, es decir, del modelo medio. Además, la reconstrucción se situará en una posición espacial parecida al modelo medio. Por tanto, si calculamos el error HPE de la reconstrucción respecto al GT3D, puede existir un error absoluto muy elevado en traslación, sobretodo en T_z. Una posible solución es realizar *Procrustes* para trasladar el modelo reconstruido a la ubicación del modelo ideal, aunque puede falsear la estimación del error debido a que se alinea también ajustando la rotación del modelo.

Puntos de *IntraFace* ideales o GT2D.

Error	Tx	Ty	Tz	Tmedia (mm)	*Roll* (º)	*Yaw* (º)	*Pitch* (º)	Media (º)
Absoluto	0,14	0,26	0,62	0,34	2,09	0,19	1,92	1,40
GT - HPE zeroed					1,59	0,71	2,00	1,43
GT zeroed - HPE zeroed	0,13	0,07	0,51	0,23	0,10	0,19	0,41	0,23

Tabla 2.5. Error HPE medio para todos los usuarios con *Procrustes*. Puntos de *IntraFace* GT2D.

El error absoluto es de partida, haciendo el análisis de *Procrustes*, similar al obtenido por medio de la modificación *zeroed*. El error modificado en **GT zeroed – HPE zeroed** tiene una valoración del error en HPE más objetiva que aquellas que comparan al menos uno de los modelos con un sistema de referencia absoluto.

Las siguientes tablas reflejan el error en traslación debido al desajuste, pero se puede estimar de manera correcta el error cuando ambos modelos se referencian a una posición nula. Las tablas se corresponden al error en HPE en traslación y rotación para todos los usuarios de manera individual y el error promedio que se obtiene con todos los resultados. Se calcula este error tanto para puntos de *IntraFace* GT2D y promediados.

	Absoluto			GT zeroed - HPE zeroed		
USUARIO	Tx	Ty	Tz	Tx	Ty	Tz
1	6,92	149,90	174,76	8,10	5,05	12,43
2	13,54	144,24	48,05	13,65	4,73	5,72
3	8,07	139,02	41,66	8,15	7,60	12,97
4	6,30	140,89	55,70	6,13	4,30	9,48
5	8,60	151,05	128,42	8,43	7,42	15,03
6	18,73	146,75	83,65	19,38	3,96	6,16
7	6,91	139,42	96,72	6,87	3,95	10,44
8	11,99	166,09	128,50	10,47	5,72	11,80
9	19,36	149,49	132,10	19,38	6,66	7,12
10	19,57	153,79	65,02	19,38	9,91	14,54
MEDIA	**12,00**	**148,06**	**95,46**	**11,99**	**5,93**	**10,57**
MEDIA T.	**85,17**			**9,50**		

Tabla 2.6. Error HPE en traslación. Puntos de *IntraFace* GT2D.

	Absoluto			GT - HPE zeroed			GT zeroed - HPE zeroed		
USUARIO	*Roll* (º)	*Yaw* (º)	*Pitch* (º)	*Roll* (º)	*Yaw* (º)	*Pitch* (º)	*Roll* (º)	*Yaw* (º)	*Pitch* (º)
1	4,70	0,72	4,73	4,70	0,72	4,73	0,04	0,03	0,07
2	2,83	0,29	5,54	2,84	0,29	5,54	0,06	0,07	0,18
3	5,32	0,04	5,60	5,33	0,03	5,60	0,01	0,03	0,05
4	3,70	0,30	4,07	3,70	0,30	4,07	0,15	0,29	0,05
5	5,78	0,51	4,69	5,78	0,50	4,69	0,09	0,08	0,06
6	2,46	0,42	2,02	2,45	0,43	2,02	0,09	0,08	0,04
7	4,76	0,52	4,57	4,76	0,51	4,57	0,12	0,16	0,04
8	3,66	0,37	3,99	3,66	0,36	3,99	0,09	0,12	0,04
9	2,83	2,38	2,96	2,83	2,36	2,96	0,17	0,12	0,08
10	4,84	0,66	4,48	4,83	0,67	4,48	0,19	0,03	0,15
MEDIA	**4,09**	**0,62**	**4,27**	**4,09**	**0,62**	**4,27**	**0,10**	**0,10**	**0,08**
MEDIA T.	**2,99**			**2,99**			**0,09**		

Tabla 2.7. Error HPE en rotación. Puntos de *IntraFace* GT2D.

Puntos de *IntraFace* promediados.

	Absoluto			GT zeroed - HPE zeroed		
USUARIO	Tx	Ty	Tz	Tx	Ty	Tz
1	7,24	149,40	172,54	8,20	5,20	15,93
2	13,33	144,18	50,80	13,41	4,71	6,60
3	7,91	138,63	39,63	8,04	7,64	13,92
4	6,27	140,63	52,00	6,03	4,39	10,59
5	8,59	150,70	129,40	8,33	7,36	17,59
6	18,64	146,22	78,15	19,28	4,12	6,52
7	6,61	138,99	96,30	6,67	3,74	9,69
8	12,06	166,08	124,59	10,55	5,88	13,92
9	19,54	149,60	126,93	19,60	6,87	10,86
10	19,27	153,85	61,75	19,20	9,75	13,73
MEDIA	**11,94**	**147,83**	**93,21**	**11,93**	**5,96**	**11,94**
MEDIA T.	**84,33**			**9,94**		

Tabla 2.8. Error HPE en traslación. Puntos de *IntraFace* promediados.

	Absoluto			GT - HPE zeroed			GT zeroed - HPE zeroed		
USUARIO	*Roll* (º)	*Yaw* (º)	*Pitch* (º)	*Roll* (º)	*Yaw* (º)	*Pitch* (º)	*Roll* (º)	*Yaw* (º)	*Pitch* (º)
1	5,26	1,03	4,85	5,26	1,03	4,85	0,84	0,58	0,31
2	2,87	0,46	5,69	2,88	0,46	5,69	0,52	0,44	0,54
3	4,51	0,63	5,81	4,51	0,62	5,81	1,13	0,63	0,25
4	3,30	0,29	4,25	3,30	0,29	4,25	0,87	0,29	0,34
5	5,71	0,85	4,58	5,86	0,79	4,60	0,53	0,52	0,23
6	2,14	1,03	2,04	2,17	1,03	2,04	0,47	0,90	0,72
7	4,46	0,80	4,44	4,49	0,78	4,44	0,59	0,54	0,23
8	3,56	0,45	4,05	3,56	0,43	4,04	0,77	0,60	0,20
9	3,14	2,01	3,24	3,14	1,99	3,24	0,50	1,11	0,43
10	4,83	1,50	5,04	4,85	1,51	5,03	0,91	0,99	0,48
MEDIA	**3,98**	**0,90**	**4,40**	**4,00**	**0,89**	**4,40**	**0,71**	**0,66**	**0,37**
MEDIA T.	**3,09**			**3,10**			**0,58**		

Tabla 2.9. Error HPE en rotación. Puntos de *IntraFace* promediados.

Las tablas correspondientes al error en traslación reproducen en ambos casos el desajuste en la localización espacial entre los modelos de cabeza GT3D y reconstruido. Este error es importante, pero siempre será menos relevante que el error en rotación, ya que la cabeza media que se emplea para realizar POSIT adopta una posición completamente frontal, así como la posición de la cabeza a reconstruir en el primer fotograma del vídeo por estar generados de manera sintética y de forma bastante controlada. Si realizásemos una corrección espacial a la cabeza media llevándola a la posición del GT3D o de manera inversa, se reduciría enormemente el error en traslación absoluto.

Comparando los resultados obtenidos mediante puntos de *IntraFace* promediados y GT2D, vemos que el error en traslación es idéntico y únicamente notamos diferencias en el error en rotación, donde nos situamos en el apartado de **GT zeroed – HPE zeroed**, 0,09 frente a 0,58. En cualquier caso, error en media menor a un grado, por lo que concluimos que el algoritmo es robusto manejando el error introducido por la deriva en la detección de los puntos de *IntraFace*.

2.4. Valoración de los resultados obtenidos. Base de datos real.

2.4.1. MSE, error cuadrático medio.

Puntos de *IntraFace* ideales o GT2D.

Previamente, debemos tener en cuenta unas pocas consideraciones respecto al cálculo del MSE cuando se trabaja con la base de datos real, debido a unas pequeñas diferencias encontradas respecto a los modelos utilizados que son comparados para calcular el error, respecto a los modelos con los que trabajábamos en la base de datos sintética.

El MSE se calcula siempre respecto a un modelo tridimensional de referencia perfectamente caracterizado, nuestro GT3D para cada modelo. El problema existe cuando los puntos correspondientes al GT3D son distintos a los empleados en la reconstrucción del modelo. Si no existe esta correspondencia, no se puede realizar de forma adecuada el análisis de *Procrustes* para alinear los dos modelos y, por tanto, no es factible obtener un MSE válido.

La solución propuesta para obtener un MSE aproximado, desde la base que no hay correspondencia de puntos entre los dos modelos, pues el modelo reconstruido está formado por los 43 puntos de *IntraFace* y el GT3D de los modelos de la base de datos real están formados por 54 puntos que no son exactamente los puntos de *IntraFace*, es realizar una selección de puntos en el GT3D muy similares a aquellos puntos detectados por *IntraFace*.

Una vez hecha esta selección, se reasignan los puntos en los dos modelos para tener una correspondencia numérica en la distribución de los puntos. De esta forma, es posible realizar una aproximación al MSE, que nos sirve a su vez durante el proceso de reconstrucción decidir si aceptar el modelo reconstruido o descartarlo, si este valor es suficientemente alto comparado con el MSE entre el modelo GT3D y la cabeza media.

Se realiza un diezmado de puntos tanto en el GT3D como en los puntos de *IntraFace*. En las siguientes figuras se muestran en rojo los puntos seleccionados y en azul, los puntos descartados.

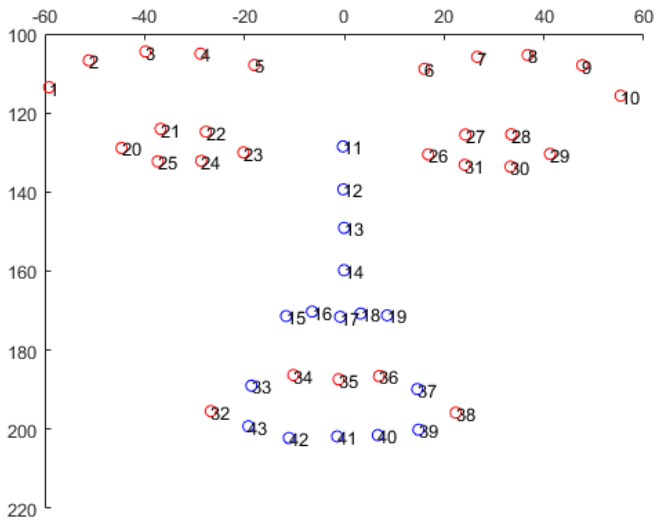


Figura 2.12. Los puntos rojos hacen referencia a los puntos de *IntraFace* seleccionados.

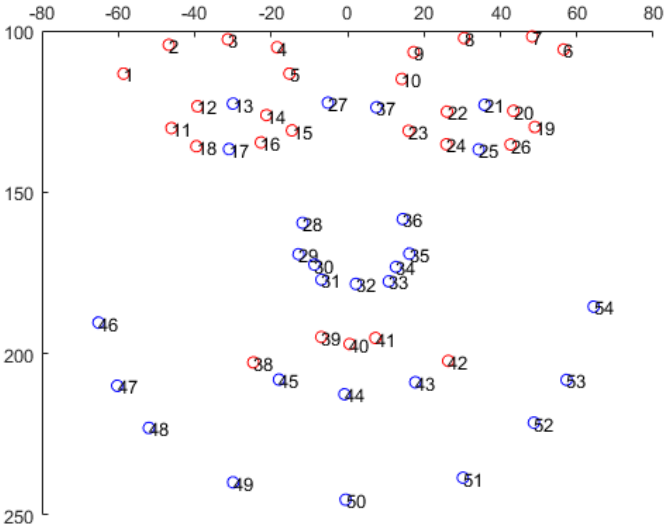


Figura 2.13. Los puntos rojos hacen referencia a los puntos GT3D seleccionados de un modelo real.

El resultado de este diezmado es la obtención de una selección con un total de 27 puntos en ambos modelos.

A continuación, se reordenan para crear una correspondencia numérica entre ambos modelos, de esta forma, es posible calcular un MSE aproximado haciendo uso del análisis de *Procrustes* cuando hay cierta equivalencia, teniendo en cuenta que la aproximación no llega a ser perfecta porque el modelo GT3D no son realmente puntos de *IntraFace*. La diferencia en la localización de los puntos es visible en las siguientes gráficas.

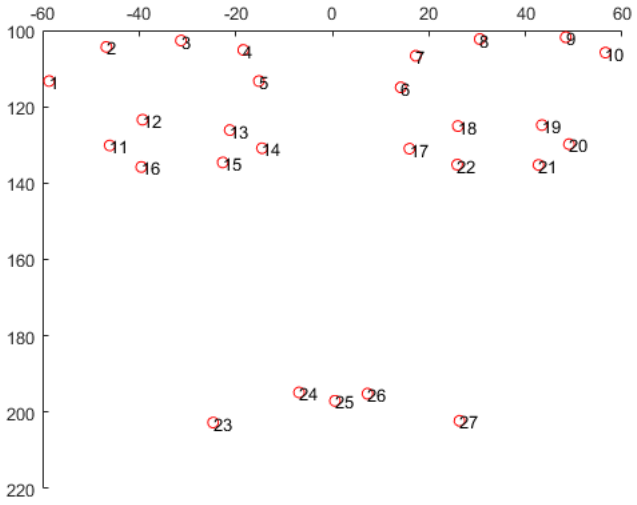


Figura 2.14. Reordenación de los puntos seleccionados en el GT3D.

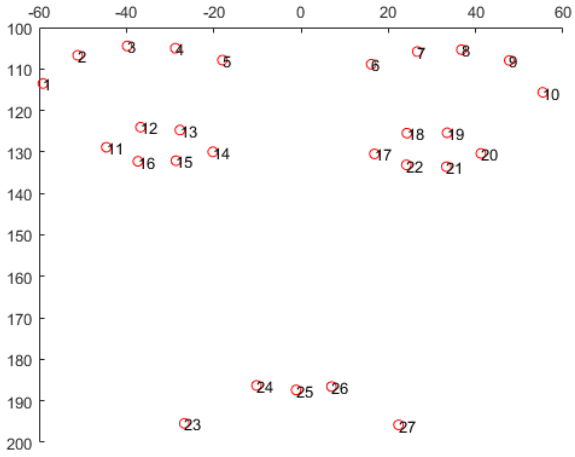


Figura 2.15. Reordenación de los puntos seleccionados en un modelo reconstruido a partir de los puntos de *IntraFace*.

Ya que el MSE es aproximado, se adjuntará a su vez el MSE obtenido al realizar un análisis de *Procrustes* con la cabeza media, caracterizada mediante puntos de *IntraFace*, y el modelo GT3D de cada usuario, caracterizado por unos puntos que no están relacionados con *IntraFace*. En ambos modelos, se realizará el diezmado y reordenación de puntos correspondientes. Entonces, cuando se compare el modelo reconstruido utilizando puntos de *IntraFace* con el GT3D, un MSE válido debe ser, como máximo, muy semejante al MSE de la cabeza media.

En este procedimiento, no es posible obtener los resultados de MSE ni HPE por la limitación de emplear un modelo de referencia, la cabeza media, caracterizado únicamente con puntos de *IntraFace*. El GT2D está formado por un conjunto de 54 puntos para cada fotograma y no es posible realizar POSIT por la misma razón encontrada en el análisis de *Procrustes*: distinto número de puntos a comparar y falta de correspondencia entre ellos.

Puntos de *IntraFace* promediados.

En este apartado, solo se mostrarán los resultados más relevantes, aquellos obtenidos empleando puntos de *IntraFace* promediados. No se incluyen los resultados para puntos de *IntraFace* sin promediar porque no presentan diferencias apreciables respecto a los promediados, empleando este método.

USUARIO	IT.	MH - GT3D	1st IT. - GT3D	Nth IT. - GT3D	DIFF.	MIN	MAX
1	2	6,05	6,09	6,09	0,00	0,46	12,63
2	4	7,08	6,06	6,01	0,05	0,58	15,79
3	2	6,94	5,59	5,59	0,00	0,59	14,14
4	6	6,69	5,22	5,21	0,01	0,32	11,13
5	2	6,20	4,54	4,54	0,00	1,92	7,38
6	5	6,25	5,33	5,28	0,06	0,40	11,80
7	2	6,24	4,86	4,86	0,00	0,53	12,63
8	2	7,35	5,39	5,39	0,00	0,64	12,10
9	2	6,88	4,75	4,75	0,00	1,57	8,87
10	4	8,45	6,54	6,48	0,06	1,38	15,28
MEDIA	**3**	**6,81**	**5,44**	**5,42**	**0,02**	**0,84**	**12,18**

Tabla 2.10. Resultados para los 10 usuarios empleando los puntos característicos de *IntraFace* promediados.

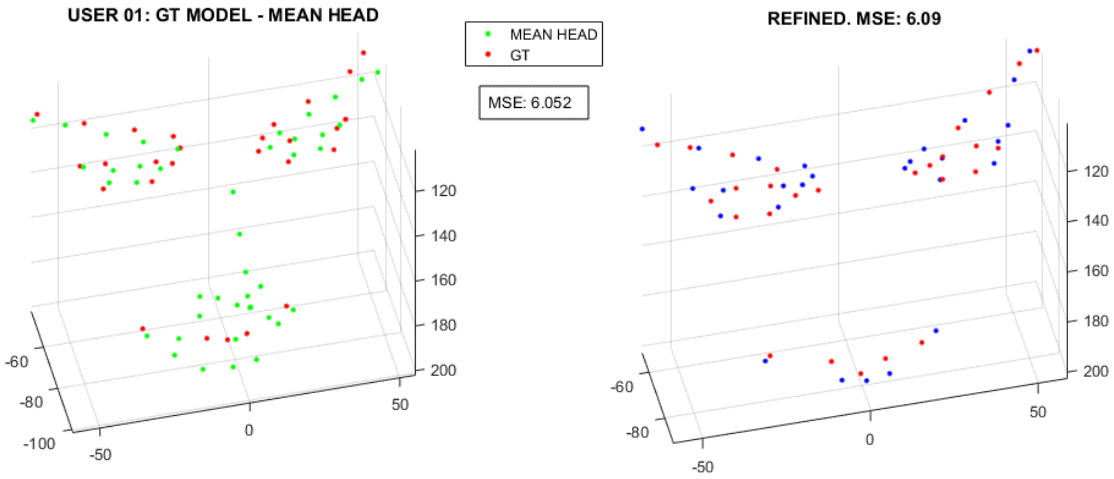


Figura 2.16. MSE MH - GT3D y Nth IT. - GT3D para el usuario 1.

La primera diferencia destacable respecto a los resultados obtenidos con los usuarios de la base de datos sintética es la rapidez en este caso de convergencia en el proceso iterativo: en media, solo se necesitan 3 iteraciones para obtener un modelo reconstruido refinado. El modelo refinado apenas difiere de aquel obtenido en la primera reconstrucción, por lo que podemos asumir dos motivos: indeterminación en el grado de mejora del modelo reconstruido debido a la aproximación de puntos en los modelos empleados al realizar el análisis de *Procrustes*, o debido a la variabilidad en los puntos detectados. Fenómeno más notorio en la base de datos real debido a las oclusiones observadas en los vídeos.

Las oclusiones se producen cuando hay cambios de expresión en las caras de las cabezas. Este fenómeno es común en todos los modelos reales, primordialmente por la necesidad de pestañear. También pueden producirse oclusiones en los puntos relativos a las cejas o los labios, aunque en mucha menor proporción que los ojos. Estas oclusiones añaden cierta componente de error porque el algoritmo de detección de *IntraFace* debe estimar los puntos tanto el cambio generado por la rotación de la cabeza como el cambio generado, por ejemplo, al pestañear.

Figura 2.17. Oclusiones de puntos observadas entre dos fotogramas en el usuario 10.

Este fenómeno ocurre en todos los modelos. Siempre hay pequeños cambios faciales intrínsecos que nos alejan de la situación ideal de los modelos de la base de datos sintética, donde sus expresiones faciales son completamente invariables entre fotogramas.

Los resultados son aproximados debido a la aproximación, pero podemos considerar las reconstrucciones totalmente válidas porque, en media, el **MSE Nth IT. - GT3D** siempre se sitúa por debajo al **MSE MH - GT3D**. Para una reconstrucción perfecta, el **MSE Nth IT. - GT3D**, idealmente, debe tender a cero.

2.4.2. Error en HPE.

Puntos de *IntraFace* promediados.

El error HPE es estimable siempre que existan correspondencias, porque el algoritmo que calcula el HPE para ser posteriormente comparados con el *ground truth* es POSIT, en cualquier caso. De forma que en este apartado se adjuntan los resultados obtenidos para los puntos de *IntraFace* promediados.

	Absoluto			GT zeroed - HPE zeroed		
USUARIO	Tx	Ty	Tz	Tx	Ty	Tz
1	220,77	19,40	200,19	12,45	1,21	4,06
2	212,15	20,74	222,96	18,64	3,10	9,54
3	200,71	20,93	215,61	12,71	1,88	9,77
4	219,40	15,60	262,82	9,67	1,42	7,25
5	211,34	30,19	251,85	14,08	2,45	11,44
6	204,59	32,20	183,54	25,39	2,91	5,74
7	240,79	13,36	283,35	17,57	2,26	4,82
8	222,66	32,34	230,04	14,00	3,80	6,89
9	216,15	17,03	264,78	28,15	5,92	8,44
10	183,55	46,74	178,40	23,24	8,61	7,07
MEDIA	**213,21**	**24,85**	**229,36**	**17,59**	**3,36**	**7,50**
MEDIA T.	**155,81**			**9,48**		

Tabla 2.11. Error HPE en traslación. Puntos de *IntraFace* promediados.

	Absoluto			GT - HPE zeroed			GT zeroed - HPE zeroed		
USUARIO	*Roll* (º)	*Yaw* (º)	*Pitch* (º)	*Roll* (º)	*Yaw* (º)	*Pitch* (º)	*Roll* (º)	Yaw (º)	Pitch (º)
1	2,78	1,73	1,27	2,32	1,69	1,18	1,66	1,57	1,24
2	1,15	1,28	5,05	1,13	1,26	5,05	1,49	1,44	1,75
3	1,81	1,09	3,18	1,65	1,08	3,16	1,12	1,10	1,26
4	1,36	1,09	2,69	1,30	1,07	2,69	0,74	1,11	1,42
5	2,85	1,02	1,89	3,27	0,98	1,79	0,90	1,10	2,10
6	3,55	0,91	1,54	3,56	0,91	1,54	2,84	1,02	1,98
7	2,78	2,11	1,88	2,78	2,17	1,89	1,83	1,78	2,22
8	1,76	0,91	2,08	1,86	0,96	2,11	2,24	0,80	1,59
9	1,42	3,99	3,40	1,42	3,91	3,31	0,87	1,61	2,00
10	2,36	1,80	4,68	2,37	1,78	4,67	0,91	1,54	2,39
MEDIA	**2,18**	**1,59**	**2,77**	**2,17**	**1,58**	**2,74**	**1,46**	**1,31**	**1,79**
MEDIA T.	**2,18**			**2,16**			**1,52**		

Tabla 2.12. Error HPE en rotación. Puntos de *IntraFace* promediados.

Si comparamos estos resultados a los homólogos en la base de datos sintética, vemos que son similares. En este caso, los resultados en traslación son mejores, pero tenemos obtenemos un error 1º en rotación, aunque el error absoluto disminuye un 1º respecto a la otra base de datos.

3. Reconstrucción tridimensional de un modelo de cabeza mediante puntos de *IntraFace*. Estimación de las poses de la cámara por medio de la Matriz Fundamental.

3.1. Introducción.

El algoritmo propuesto en el capítulo anterior, se basa principalmente en la extracción de las poses de la cámara a partir de la estimación de la pose de la cabeza en cada fotograma por medio de un algoritmo externo denominado POSIT. Posteriormente, se itera el proceso tantas veces como se requiera para considerar que el modelo reconstruido ha convergido y las poses obtenidas son óptimas, y no encontramos diferencias apreciables en estas poses al continuar refinando el modelo.

La limitación principal del anterior algoritmo es el requerimiento de un modelo tridimensional necesario y adecuado para estimar las poses de la cabeza. Se fundamenta en realizar una transformación a este modelo para que la proyección sobre los puntos de cada fotograma sea coincidente. La transformación que ha aplicado al modelo para minimizar el error en la proyección es la estimación de las poses de la cabeza.

Nuestro modelo de referencia es la cabeza media de BFM que consigue obtener unos resultados bastante razonables, aunque tiene la limitación que únicamente funciona para puntos detectados mediante *IntraFace*, por estar definida mediante estos puntos y no otros. En resumen, siempre existirá una dependencia con un algoritmo externo como es POSIT y con el modelo de referencia limitado a un determinado número y tipo de puntos.

En este apartado, se sugiere un método para calcular las poses de la cámara sin emplear un modelo de referencia tridimensional. Únicamente se empleará la diferencia que existe entre la correspondencia de puntos, que pueden ser arbitrarios, entre dos fotogramas distintos para calcular directamente las poses de la cámara, aunque en este capítulo se trabajarán con los puntos de *IntraFace*.

La relación que existe entre los puntos de dos fotogramas distintos se describe por medio de la Matriz Fundamental.

Para explicar el proceso de obtención de la Matriz Fundamental, es necesario introducir el modelo de geometría epipolar que modela este concepto y presenta la base fundamental de otros conceptos relacionados.

3.2. Geometría epipolar.

3.2.1. Definición.

La geometría epipolar es la geometría proyectiva intrínseca entre dos puntos de vista. Es independiente de la estructura de la escena, y sólo depende de los parámetros internos de la cámara y la pose relativa. Es la geometría de la visión estereoscópica [11].

Cuando dos cámaras observan una escena 3D desde dos posiciones distintas, hay una serie de relaciones geométricas entre los puntos 3D y sus proyecciones sobre las imágenes en 2D que conducen a definir una relación entre los puntos de imagen.

Estas relaciones se derivan en base al supuesto de que las cámaras pueden ser aproximadas por el modelo de la cámara *pinhole* o estenopeica, es decir, suponiendo una profundidad de campo infinita, donde un punto en el espacio se proyecta en forma de rayo dando lugar a un único punto en una imagen del sensor.

Suponemos que dos cámaras estenopeicas buscan un punto P. En las cámaras reales, el plano de la imagen está en realidad detrás del centro focal, y produce una imagen que es simétrica respecto al centro focal de la lente. Aquí, sin embargo, el problema se simplifica mediante la colocación de un plano virtual de imagen enfrente del centro óptico de cada lente de la cámara para producir una imagen no transformada por simetría.

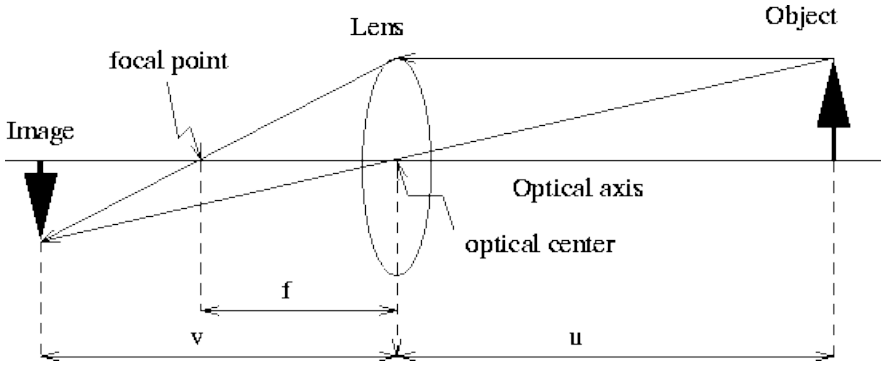


Figura 3.1 Formación de una imagen en el sensor de una cámara. Los puntos se proyectan detrás del punto focal formando una imagen simétrica.

Cada cámara captura una imagen bidimensional de una escena tridimensional. Esta conversión de 3D a 2D se denomina proyección en perspectiva. Para modelar esta operación, la proyección de los rayos que emanan de la cámara, pasan por su centro focal correspondiendo a un solo punto en la imagen suponiendo siempre un modelo de cámara estenopeica para ignorar efectos ópticos como el círculo de confusión presentes en las lentes reales.

La siguiente figura representa de manera fundamental los conceptos básicos de la geometría epipolar y las relaciones entre estos conceptos: epipolos, líneas epipolares y plano epipolar.

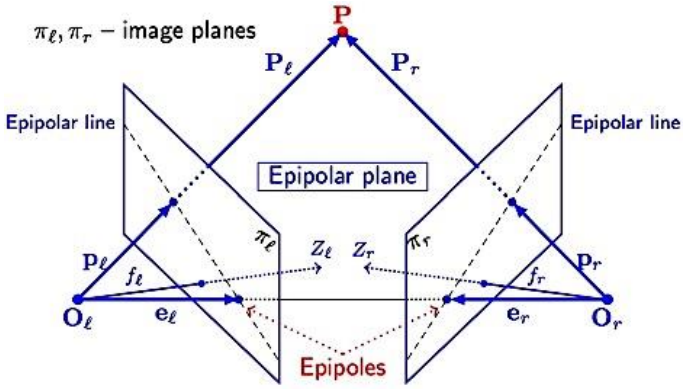


Figura 3.2. Representación fundamental de los conceptos de la geometría epipolar.

O_L y O_R representan los centros de simetría de las dos lentes de las cámaras. P representa el punto de interés en ambas cámaras. Puntos P_L y P_R son las proyecciones de punto P sobre los planos de imagen π_L y π_R.

3.2.2. Punto epipolar o epipolo.

Si los centros ópticos de las lentes de las cámaras están bien definidos, cada centro proyecta un punto distinto en el plano de la imagen de la otra cámara. Estos dos puntos de imagen se denotan en la figura anterior por e_L y e_R y se llaman epipolos o puntos epipolares.

Ambos epipolos en sus respectivos planos de imagen y los dos centros ópticos, O_L y O_R, se encuentran en una sola línea 3D.

3.2.3. Línea epipolar.

Por ejemplo, la línea O_L - P es vista por la cámara izquierda como un punto, porque está directamente en línea con el centro óptico de la cámara. Sin embargo, la cámara derecha ve esta línea como una línea en su plano de la imagen. Lo mismo ocurre en el caso contrario: la línea O_R - P es vista por la cámara derecha como punto, pero como una línea en el plano de imagen de la cámara derecha. Estas líneas (e_r – P_r y e_l - P_l) se denominan líneas epipolares.

Una línea epipolar es una función de la posición del punto P en el espacio 3D, es decir, como la variación de P genera un conjunto de líneas epipolares en ambas imágenes. Puesto que la línea 3D O_L - P pasa por el centro óptico de la lente O_L, la línea epipolar correspondiente en la imagen de la derecha tiene que pasar por el epipolo e_r (respectivamente para las líneas epipolares en la imagen de la izquierda).

En resumen, todas las líneas epipolares en una imagen contienen el punto epipolar, ya que se pueden derivar de un cierto punto P.

3.2.4. Plano epipolar.

Como una visualización alternativa, consideramos que los puntos P, O_L y O_R forman un plano, llamado plano epipolar. Este plano interseca cada plano de imagen de cada cámara donde se forman las líneas epipolares. Todos los planos epipolares y líneas epipolares cruzan el epipolo, independientemente del lugar donde se encuentre P.

3.3. Matriz fundamental.

3.3.1. Definición.

Como se ha visto, la relación que existe entre los puntos de dos imágenes está gobernada por la geometría epipolar. Cuando dos cámaras observan una escena desde dos puntos de vista distintos, hay un número de relaciones geométricas entre los puntos en el espacio y sus proyecciones en las imágenes, reflejados en los puntos de las imágenes.

La Matriz Fundamental es una relación entre dos imágenes de la misma escena que describe donde puede ocurrir la proyección de los puntos de una escena en ambas imágenes. Teniendo en cuenta que la proyección de un punto de la escena en una de las imágenes junto con el punto correspondiente en la otra imagen (denominado a partir de ahora *inlier*) está limitado por una línea, permite conocer a su vez la detección de correspondencias erróneas (*outliers*).

La matriz fundamental, F, encapsula esta geometría intrínseca. Es una matriz de 3 x 3 de rango 2, cuyo kernel define el epipolo. Un punto en el espacio, X, se forma en la imagen como x en la primera vista, y x' en la segunda.

Fx describe una línea (línea epipolar) en la que debe situarse el correspondiente punto x' en la otra imagen. Entonces los puntos de imagen satisfacen la relación $x'^T F x = 0$ [11].

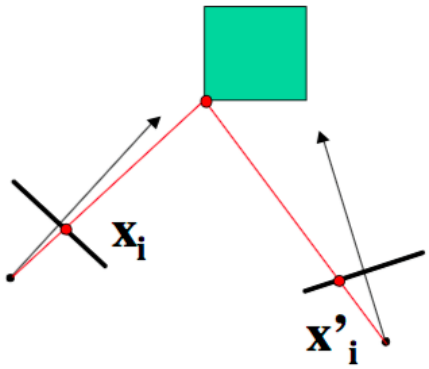


Figura 3.3. Proyecciones en distintos planos para un mismo punto *x*.

La Matriz Fundamental se puede estimar con al menos ocho correspondencias de puntos. El método más empleado es el algoritmo de 8 puntos normalizado.

3.3.2. Algoritmo de 8 puntos normalizado.

La Matriz Fundamental se define por la ecuación $X'^T F X = 0$ (1) para cualquier pareja o conjunto de puntos correspondientes, $X' \leftrightarrow X$, en dos imágenes.

Dado un suficiente número de puntos correspondientes, $X'_i \leftrightarrow X_i$, al menos 8, la ecuación (1) se puede utilizar para calcular la Matriz Fundamental desconocida. En concreto, escribiendo $X = (x, y, 1)$ y $X' = (x', y', 1)$, cada punto eleva a uno el número de incógnitas en la ecuación lineal de F [13].

Los coeficientes de esta ecuación se pueden escribir fácilmente en términos de las coordenadas conocidas X y X'. Concretamente, la ecuación correspondiente a una pareja de puntos $(x, y, 1)$ y $(x', y', 1)$ será

$$xx'f_{11} + xy'f_{21} + xf_{31} + yx'f_{12} + yy'f_{22} + yf_{32} + x'f_{13} + y'f_{23} + f_{33} = 0.$$

La fila de la ecuación matriz puede representarse como un vector $(xx', xy', x, yx', yy', y, x', y', 1)$.

Desde todas las correspondencias de puntos, se obtiene un conjunto de ecuaciones lineales de la forma $Af = 0$ (2), donde f corresponde a un vector de 9 elementos que contiene las incógnitas de la matriz F, y A es la matriz ecuación. La Matriz Fundamental F, por consiguiente, es la solución del vector f y está definida en una escala desconocida. Por esta razón, y para evitar la solución trivial de f, añadimos una condición adicional $//f// = 1$, donde $//f//$ es la norma de f.

Bajo estas condiciones, es posible encontrar una solución al sistema (2) con al menos 8 correspondencias de puntos.

La solución lineal, dados unas correspondencias de puntos $X'_i \leftrightarrow X_i$, se resuelven las ecuaciones $X_i'^T F X_i = 0$ para encontrar F. La solución es el menor vector propio, f de $A^T A$, donde A es la matriz ecuación.

Las coordenadas de la imagen a veces se dan con el origen en la parte superior izquierda de la imagen, y a veces con el origen en el centro. Esto hace preguntarnos si existe una diferencia en los resultados del algoritmo de 8 puntos para el cálculo de la matriz fundamental o en qué medida es el resultado del algoritmo de 8 puntos dependiente de la elección de las coordenadas en la imagen.

Normalizar los puntos resuelve los problemas de escalado y traslación entre las distintas incógnitas de la matriz ecuación. La normalización consiste en un escalado isotrópico: trasladar los puntos a un origen definido por el centroide de éstos, de tal manera que la distancia promedio es equivalente a $\sqrt{2}$.

Emplear este método de cálculo de la matriz fundamental es válido si la correspondencia de puntos es precisa, es decir, suponiendo que todos los puntos de entrada en el cálculo de la Matriz Fundamental son *inliers*, por tanto, hay un problema en calcular la Matriz Fundamental si se parte de la premisa de que no todos los puntos detectados en la imagen corresponden a *inliers* y existe la posibilidad de encontrarnos *outliers* que rompan la relación $x'^T F x = 0$. Los *outliers* pueden estar relacionados con el error generado en la variabilidad de la propia detección de estos puntos. Para ello, existen múltiples algoritmos o técnicas que generan una línea de tendencia de *inliers* a partir de un subconjunto de puntos y descartan los *outliers*. Entre ellos se encuentran el algoritmo de RANSAC y sus variantes, como MSAC, MLESAC, LO-RANSAC, AMLESAC, GASAC, etc. Las modificaciones al algoritmo original vienen a solventar carencias en las limitaciones originales.

En el siguiente apartado, se explicará el fundamento básico de funcionamiento del algoritmo RANSAC seguido de un estudio para determinar la configuración más adecuada para nuestra aplicación.

3.3.3. Técnicas de estimación y exclusión de *outliers*. RANSAC.

RANSAC (RANdom SAmple Consensus) es un algoritmo iterativo estadístico, publicado por primera vez por Fischler y Bolles, en SRI International, 1981. Es capaz de calcular los parámetros de un modelo matemático de un conjunto de datos observados que contiene *outliers* o valores atípicos. Al ser estadístico, no produce resultados deterministas, de forma que sólo produce un resultado válido con una cierta probabilidad, en mayor medida si elevamos el número de iteraciones [14].

En nuestro caso, existe un número de puntos o datos óptimos, denominados *inliers*, rodeados por otros puntos susceptibles al ruido generado en su detección o en la variabilidad de los algoritmos de seguimiento. Estos datos se denominan *outliers* o valores atípicos.

Para entender la aplicación de este algoritmo, se ilustra un ejemplo sencillo dado un conjunto de puntos u observaciones. Asumiendo que este conjunto contiene todos los *inliers*, es decir, los puntos que se pueden insertar de forma aproximada en una línea, y los *outliers*, puntos que no se excluyen de esta línea. Si se utiliza el método de mínimos cuadrados ordinarios para el montaje de la línea, generalmente producirá una línea con un mal ajuste de los *inliers*. La razón es que se monta de forma óptima a todos los puntos sin discriminar si estos siguen la línea de tendencia general, es decir, incluyendo los *outliers*.

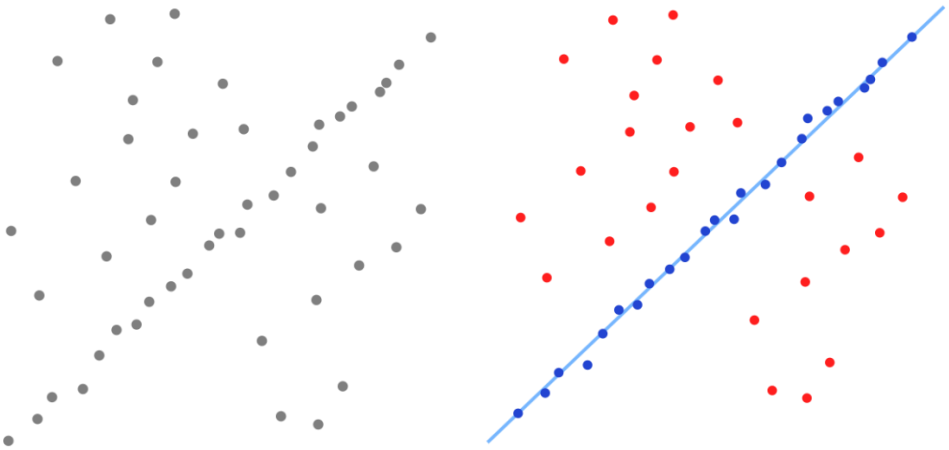


Figura 3.4. A la izquierda, conjunto de puntos donde se incluyen inliers y outliers. A la derecha, cálculo de la línea de regresión a partir de los inliers, excluyendo los outliers por medio de RANSAC.

RANSAC permite crear un modelo que sólo está calculado con los *inliers*, aunque su funcionamiento es óptimo siempre que la probabilidad de elegir sólo los *inliers* sea suficientemente alta, generalmente superior al 50% del conjunto de datos. No hay ninguna garantía de que se cumpla esta situación, sin embargo, hay una serie de parámetros del algoritmo que deben elegirse cuidadosamente para mantener el nivel de probabilidad razonablemente alta, como el intervalo de confianza de los *inliers* en el conjunto y el umbral de distancia para encontrar los *outliers*.

La entrada al algoritmo RANSAC está formado por un conjunto de valores de los datos observados, una forma de ajustar algún tipo de modelo a las observaciones y algunos parámetros de confianza. RANSAC logra su objetivo mediante la repetición de los siguientes pasos:

1. Seleccionar un subconjunto aleatorio de los datos originales llamado "*inliers* hipotéticos", para determinar los parámetros del modelo. En nuestra implementación, un subconjunto a partir de 8 puntos.

2. Resolver para los parámetros del modelo.

3. Se determina cuantos puntos de todo el conjunto se ajustan dentro del modelo, de acuerdo con algún valor de tolerancia, es decir, aquellos que se consideran como parte del conjunto de consenso.

4. Si la fracción del número de *inliers* sobre el total de puntos del conjunto excede un determinado umbral, se vuelven a estimar los parámetros del modelo usando todos los *inliers* identificados y se termina el proceso.

5. Si no se cumple el paso 4, se repiten los pasos del 1 al 4 hasta un máximo número de iteraciones.

Este procedimiento se repite un número variable de veces, produciendo o bien un modelo que es rechazado porque no hay suficientes puntos en el conjunto de consenso, o un modelo aceptado con suficientes puntos en el conjunto de consenso. En este último caso, mantenemos el modelo si su conjunto de consenso es más grande que el modelo guardado previamente.

Parámetros

El intervalo de confianza define una estimación, por parte del usuario, de un porcentaje tal que el algoritmo RANSAC seleccione solo *inliers* dentro del conjunto de datos. Parámetro denominado p. En la práctica, resulta complicado estimar el intervalo óptimo del máximo número de *inliers*, ya que es dependiente del conjunto de datos de entrada al algoritmo en ese momento, fenómeno que se comentará posteriormente en el caso de los puntos de *IntraFace*.

El número de puntos que se eligen del subconjunto de entrada se denomina n. En nuestra implementación este valor es 8 o superior.

La probabilidad de elegir un *inlier* cada vez que se selecciona un solo punto se denomina r.
Por tanto, r^8 es la probabilidad de que los 8 puntos elegidos sean *inliers*.

$1 - r^8$ es la probabilidad de que al menos uno de los 8 puntos es un *outlier*.

Esta probabilidad a la potencia de k es la probabilidad de que el algoritmo nunca seleccione un conjunto de 8 puntos donde todos son *inliers* y esto debe ser el mismo que $1 - p$, donde k es el valor máximo de iteraciones del algoritmo.

$1 - p = (1 - r^8)^k$, despejando k:

$$k = \frac{log(1-p)}{log(1-r^8)}$$

Ecuación 3.1.

N se define a su vez como el número máximo de iteraciones para encontrar los *outliers*. En cada iteración, RANSAC actualiza este valor dado por el mínimo entre el valor actual de N y el valor de k.

$$N = min\left(N, \frac{\log(1-p)}{\log(1-r^8)}\right)$$

Ecuación 3.2.

La implementación de MATLAB calcula r como:

$$\sum_{i}^{N} sgn(d(u_i, v_i), t)/N$$

Ecuación 3.3.

donde, $sgn(a, b) = 1$ si a ≤ b. En otro caso, 0.

La distancia geométrica entre puntos correspondientes en cada imagen se denomina $d(u_i, v_i)$.
El umbral de distancia para encontrar los *outliers* se corresponde al parámetro t.

En la práctica, podemos introducir como parámetros de entrada el intervalo de confianza, el número máximo de iteraciones y el umbral de distancia para encontrar los *outiliers*, p, N y t.

A partir de la ecuación (3), observamos que, si r disminuye, condicionado por la elección de un umbral de distancia de *outliers* muy pequeña, tiene un impacto en el denominador de la ecuación 3.1, elevando su valor y, por tanto, aumentando ese término en la ecuación 3.2. Por tanto, cuando se actualice el número de iteraciones necesarias para establecer la condición de parada del algoritmo, k será muy elevado y se escogerá siempre el límite máximo de iteraciones.

Esta situación tiene un impacto negativo en el coste computacional, ya que aumenta el tiempo de cálculo, pero crece la probabilidad de que la Matriz Fundamental esté mejor calculada.

Por tanto, se debe alcanzar un compromiso en el número de iteraciones para aumentar las posibilidades de obtener un resultado óptimo sin aumentar en exceso el tiempo de cálculo. Está solución de compromiso es la gran limitación del algoritmo, ya que partimos de un algoritmo no determinista con resultados no idénticos entre ejecuciones debido a la selección aleatoria del subconjunto de puntos.

Es recomendable conocer a priori la naturaleza de la presencia de *outliers* para establecer la configuración óptima entre coste computacional y validez del resultado.

3.3.4. Variabilidad de los puntos de *IntraFace*. Ruido de detección.

La dificultad de este método radica principalmente en la discriminación de *inliers*, debido a que la detección de *IntraFace* genera variabilidad en la disposición de los puntos y no es adecuado realizar el cálculo de la Matriz Fundamental aplicando directamente el método de 8 puntos normalizado, ya que sería correcto solo en el caso que los puntos coincidieran perfectamente, como puede ser el caso de los puntos de *IntraFace* que conforman el *ground truth*, donde la correspondencia de puntos no da lugar a derivas entre imágenes.

MSE de *IntraFace.*

Se ha realizado un estudio previo para comprobar la variabilidad de los puntos de *IntraFace* detectados en las distintas imágenes de la base de datos sintética, puesto que se conoce con bastante precisión los puntos de *IntraFace* correspondientes de manera ideal mediante el *ground truth* 2D, de forma que se puede calcular la distancia euclídea entre estos puntos detectados y los ideales para cada fotograma.

Obtenemos una visión global de los puntos problemáticos, aquellos donde la distancia entre el punto detectado y el ideal es muy grande, indicándonos que pueden tratarse de posibles *outliers*, desembocando en un error acumulado en el cálculo de la Matriz Fundamental. Además, se puede conocer si estos *outliers* presentan un patrón entre los diversos fotogramas y de qué magnitud tiene este error. Las conclusiones obtenidas en este apartado permiten realizar una configuración óptima si se escoge el método RANSAC para excluir *outliers*.

Se han evaluado los tipos de vídeos que se emplean para la reconstrucción mediante SfM, tanto para la base de datos sintética como para la base de datos real, es decir, aquellos con movimiento en *yaw* y en *pitch* puros (sin traslación y sin combinación de otros tipos de rotación). Se evalúan los 300 fotogramas.

A continuación, se representa para el décimo usuario de la base de datos sintética, el movimiento en grados que efectúa la cabeza en cada fotograma para un determinado tipo de movimiento y un *boxplot* que representa la cantidad de *outliers* de cada fotograma, así como la varianza del error. La línea roja es la mediana, los límites del rectángulo azul son el percentil 25 y 75, representan la varianza y dentro del rectángulo se encuentran los *inliers*. La línea discontinua delimita a los puntos considerados *inliers* y las cruces rojas son los *outliers*.

Se representa también el error acumulado, normalizado al valor máximo, de los 43 puntos de *IntraFace* para todos los usuarios de la base de datos y su representación valorativa del error en estos puntos en el plano tridimensional de manera escalada.

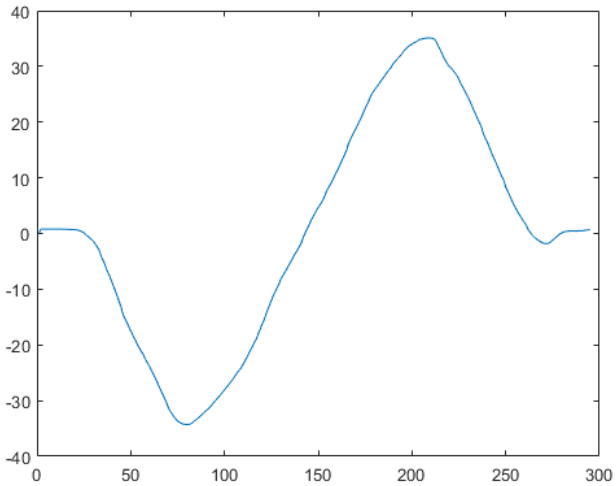


Figura 3.5. Vídeo con movimiento tipo *yaw*. Eje de abscisas representa el número de fotograma. Eje de ordenadas representa el grado de movimiento.

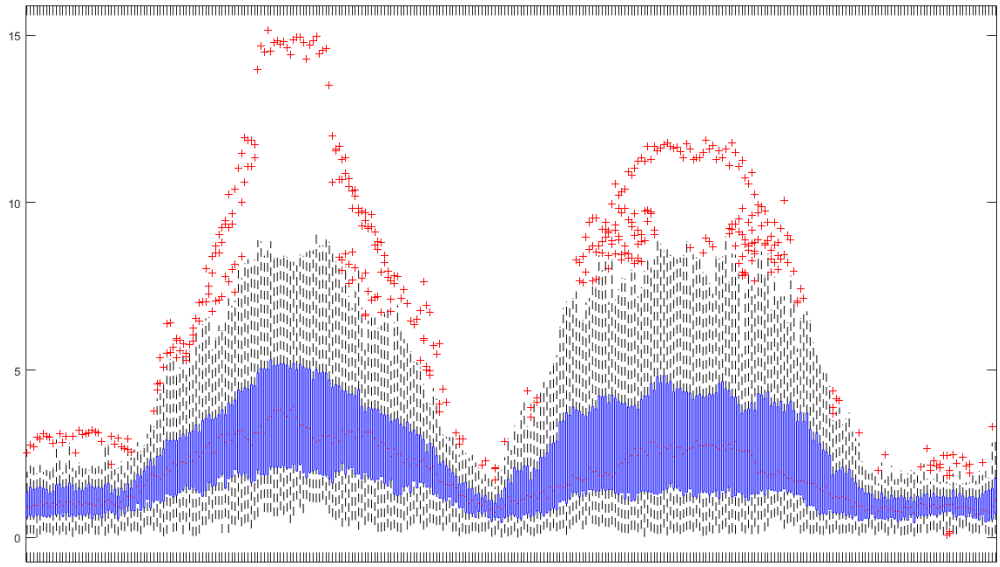


Figura 3.6. *Boxplot* para el vídeo con movimiento tipo *yaw*. Eje de abscisas representa el número de fotograma. Eje de ordenadas representa la distancia euclídea de los puntos detectados respecto al *ground truth*.

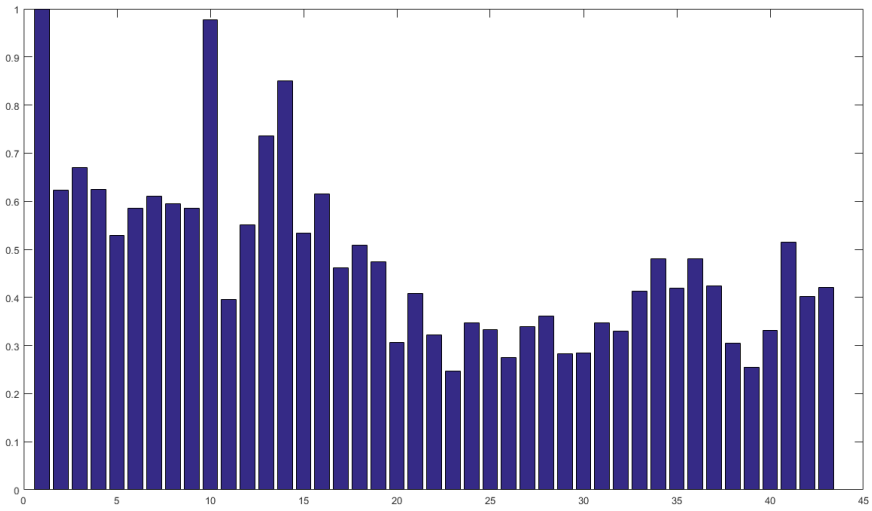


Figura 3.7. Error acumulado para los 43 puntos de *IntraFace* para todos los usuarios de la base de datos sintética con vídeo de movimiento tipo *yaw*. Se observa la diferencia de error entre los puntos.

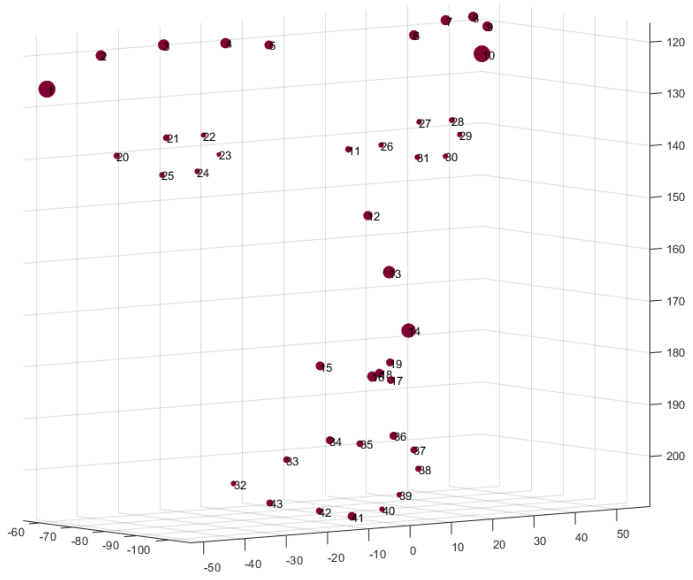


Figura 3.8. Representación visual y valorativa del error acumulado para los 43 puntos de *IntraFace* para todos los usuarios de la base de datos sintética con vídeo de movimiento tipo *yaw*. A mayor diámetro de esfera, mayor error.

A partir de observación de las gráficas presentadas, podemos obtener diversas conclusiones.

En las primeras figuras, vemos que el error es dependiente del ángulo de *yaw* y aumenta conforme el ángulo se aleja de la posición frontal, es decir, cuando el ángulo es de 0º. A partir de cierto ángulo, hay un salto abrupto en el error, donde alcanza el máximo, en torno a ±20º.

En las siguientes figuras, se observa el error acumulado, donde se puede apreciar qué puntos presentan el mayor error tras recorrer todos los fotogramas de los 10 usuarios, es decir, el promedio del error de 3000 fotogramas con movimiento tipo *yaw*.

Observando la concurrencia de los puntos considerados *outliers* en el *boxplot*, vemos que señalan a los puntos 1, 10 y 14, coincidiendo con los datos que arrojan las figuras del error acumulado. Estos puntos

son los dos que se sitúan a los extremos de la parte superior de las cejas y el que se corresponde con la punta de la nariz. Estos puntos son más susceptibles a este tipo de movimiento debido posiblemente a oclusiones, es decir, posiciones de la cabeza donde se alcanzan un ángulo lo suficientemente grande para dejar de ser detectado debido a la pérdida de visibilidad de estos puntos. La detección de *IntraFace*, posiblemente, interpola su posición a partir de la posición del resto de puntos.

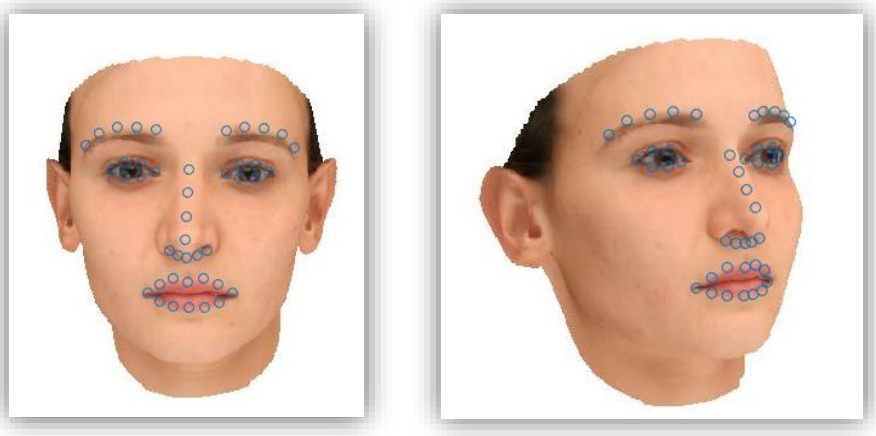


Figura 3.9. Puntos de *IntraFace* detectados en la imagen frontal del usuario 10 (izquierda) frente a los puntos detectados cuando la cabeza tiene un ángulo de 31º en movimiento tipo *yaw* (derecha). *IntraFace* marca puntos donde no existe visibilidad, como es, en este caso, el punto 10.

En el caso de seleccionar vídeos con movimiento tipo *pitch*, se obtienen las siguientes figuras.

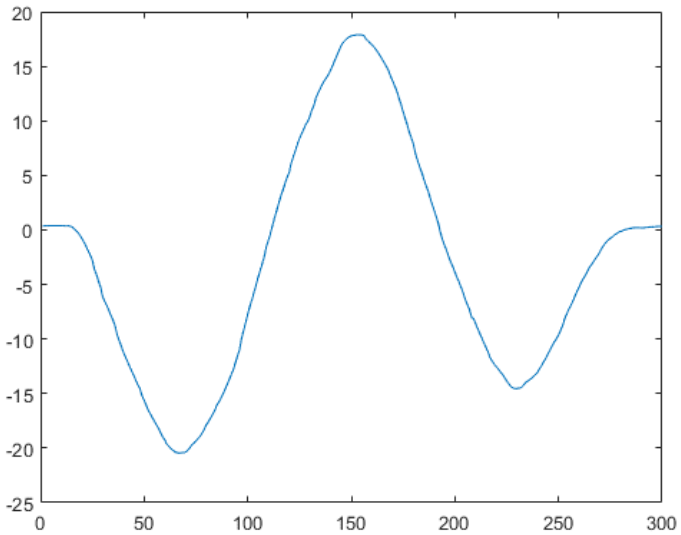


Figura 3.10. Vídeo con movimiento tipo *pitch*. Eje de abscisas representa el número de fotograma. Eje de ordenadas representa el grado de movimiento.

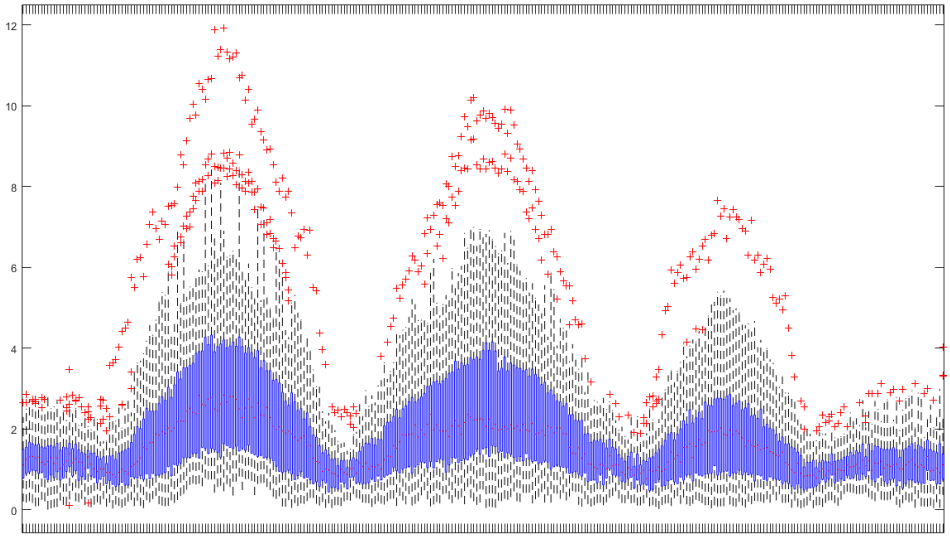


Figura 3.11. Boxplot para el vídeo con movimiento tipo *pitch*. Eje de abscisas representa el número de fotograma. Eje de ordenadas representa la distancia euclídea de los puntos detectados respecto al *ground truth*.

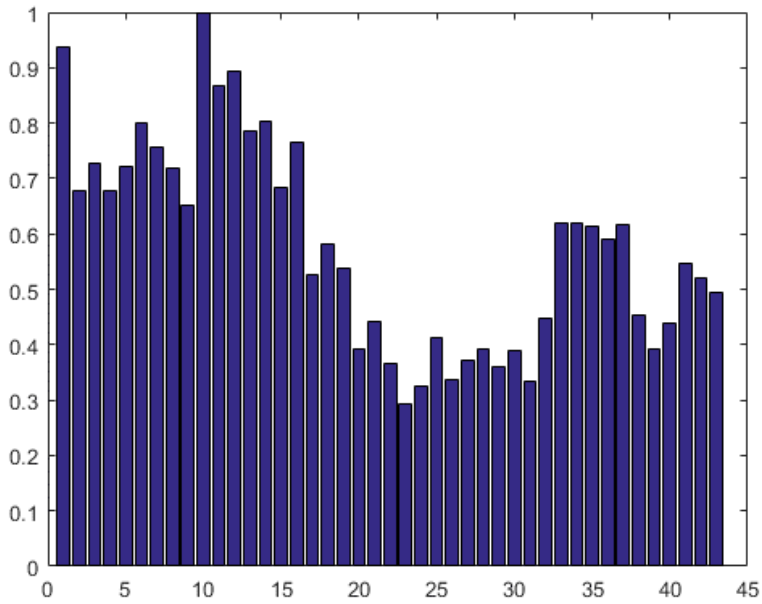


Figura 3.12. Error acumulado para los 43 puntos de *IntraFace* para todos los usuarios de la base de datos sintética con vídeo de movimiento tipo *pitch*. Se observa la diferencia de error entre los puntos.

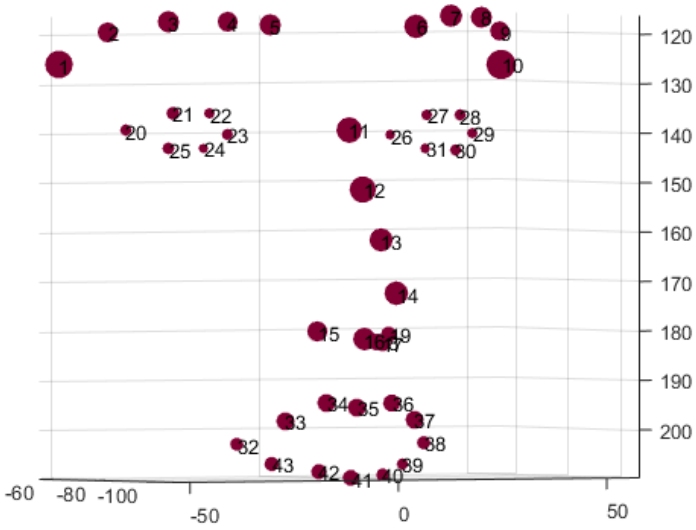


Figura 3.13. Representación visual y valorativa del error acumulado para los 43 puntos de *IntraFace* para todos los usuarios de la base de datos sintética con vídeo de movimiento tipo *pitch*. A mayor diámetro de esfera, mayor error.

Para este tipo de movimiento, el error en media es superior que el escenario anterior, pero el error alrededor de todos los puntos tiene una tendencia más uniforme. Además, la detección es más sensible a pequeñas variaciones de este tipo de movimiento, generando *outliers* de manera más prematura.

De igual manera que en el escenario anterior, destacan los puntos 1,10 y 14, aunque no de la misma forma que lo hacían en los vídeos con movimiento tipo *yaw*, pues estos picos están rodeados de otros puntos con error bastante alto.

Podemos concluir que el algoritmo de *IntraFace* es menos robusto al movimiento tipo *pitch* frente al movimiento tipo *yaw,* como corroboran todas las figuras.

Error en HPE de *IntraFace.*

Además, para obtener resultados concluyentes, se ha calculado el error que se genera en la estimación de la pose de la cabeza en función del ángulo. Para ello, se ha tomado parte de un estudio del cálculo del error en la estimación de la pose de la cabeza sobre ASM (*Active Shape Model*) de J. J. Bengoechea [9].

Las pruebas que realizó consistieron en tomar un cilindro con parámetros definidos por el GT3D. A este cilindro se ajustan los puntos del GT2D para calcular los vectores normales de cada punto, en cada fotograma. Estos vectores se comparan respecto a los obtenidos mediante los puntos procedentes de la detección ASM o AAM. El error se representa de como un histograma tridimensional, donde se puede observar en el error en función del ángulo. En el caso que existiese una dependencia entre éstos como en el caso del MSE, se podría observar una recta de regresión creciente.

Extrapolando este procedimiento para nuestro caso, con puntos de *IntraFace*, se obtiene el siguiente histograma tridimensional. El histograma tiene en cuenta todos los vídeos para todos los usuarios de la base de datos sintética, es decir, vídeos con movimiento puro tipo *yaw, pitch, roll* y combinación de los anteriores de manera aleatoria.

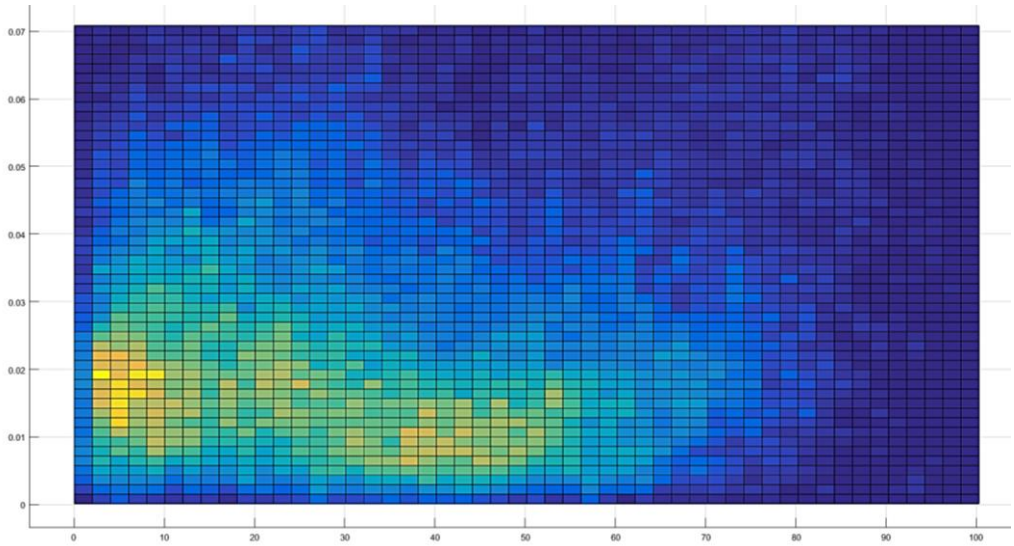


Figura 3.14. Vista cenital del histograma tridimensional del error en la estimación del Head Pose para todos los vídeos y usuarios de la base de datos sintética.

Observamos que, curiosamente, no existe dependencia del error en la detección respecto al ángulo ya que no podemos determinar una línea de regresión, el error en la estimación de la pose de la cabeza mediante puntos de *IntraFace* se mantiene constante.

Llevando estos resultados a nuestra aplicación, que hace uso de RANSAC, podemos concluir que la distancia de los *outliers* depende de la posición de la cabeza y del tipo de movimiento, pero el umbral máximo de distancia en la búsqueda de éstos se sitúa en torno a 2.5 píxeles en la vista frontal, observado en el *boxplot* del vídeo con movimiento tipo *yaw*.

El valor mínimo de error que introduce *IntraFace* respecto al GT2D es de 0.1 píxeles, por tanto, se puede buscar *outliers* a partir de este valor.

Respecto al intervalo de confianza, la probabilidad de escoger un *inlier* entre el conjunto de puntos, varía respecto al ángulo, porque el error crece y decrece de manera gradual. Depende a su vez del número de fotogramas y de si se establece un ángulo máximo en esta selección. Un valor adecuado para el intervalo de confianza puede ser 50%.

3.4. Diseño del algoritmo de reconstrucción.

3.4.1. Primera aproximación del algoritmo.

Este algoritmo, como se ha comentado, gira en torno a la estimación de las poses de la cámara por medio de la Matriz Fundamental. A priori, puede parecer una ventaja enorme tener independencia en el uso de un modelo de cabeza para estimar las poses, pues realizar HPE como se ha visto, demanda un cierto tipo, disposición y cantidad de puntos que se deben emplear en la reconstrucción de nuestro modelo y depende del modelo 3D empleado en dicha estimación. Este método nos abre un nuevo abanico de posibilidades para emplear una colección de puntos arbitraria procedente de la detección de puntos mediante otros algoritmos diferentes a *IntraFace*.

En este capítulo se evaluará el comportamiento en el cálculo de las poses de la cámara empleando la Matriz Fundamental, aunque se introducirán los puntos de *IntraFace*. Como se ha visto, *IntraFace* genera un error característico en la detección, dependiendo del tipo de movimiento aplicado a la cabeza y de su ángulo, en los puntos detectados. Es este matiz el que pondrá en duda la robustez del método y el comportamiento de algoritmos de exclusión de *outliers* que generen una fuente del error en el cálculo de las poses relativas de la cámara.

El primer planteamiento del algoritmo es una implementación propia, adaptada a los puntos de *IntraFace*, del algoritmo de reconstrucción multivista que sugiere MATLAB de manera genérica.

1. Para cada par de imágenes consecutivas, encontrar un conjunto de correspondencias de puntos. Se detectan los puntos de interés utilizando el algoritmo de *IntraFace* y se realiza el seguimiento de estos puntos mediante su algoritmo propio, combinando la detección en el fotograma actual con detecciones anteriores para mejorar la estimación.

2. Estimar la pose relativa de la vista actual. Es la orientación de la cámara y la ubicación con respecto a la imagen anterior. Se inicia un proceso que se encarga de calcular la Matriz Fundamental a partir de una serie de puntos correspondientes, localizados en dos fotogramas distintos. Con dicha matriz, se calcula las poses relativas entre ambas imágenes. Una función auxiliar puede permitirnos escoger el método de cálculo de la Matriz Fundamental, así como los parámetros asociados a cada uno, llamada *estimateRelativePose*. La familia de métodos RANSAC tienen un comportamiento estocástico debido a la naturaleza estadística del mismo. Se puede generar un bucle con una condición de salida tal que, la fracción de validación del cálculo de la Matriz Fundamental sea elevada para asegurar el correcto cálculo y aproximar en la máxima medida a un resultado determinista.

3. Transformar la pose relativa de la vista actual al sistema de coordenadas de la primera imagen de la secuencia. La Matriz Fundamental se calcula entre imágenes sucesivas. Se transforman las poses obtenidas de acuerdo al sistema de referencia del primer fotograma.

   orientación actual = orientación anterior * orientación relativa actual
   localización actual = localización anterior + localización relativa actual * orientación anterior

4. Almacenar los atributos de la vista actual: pose de la cámara y de los puntos de imagen.

5. Almacenar las correspondencias *inlier* entre la antigua y la vista actual. En este caso, todos los puntos son considerados *inlier* porque se ha realizado *tracking* y no *matching*.

6. Encontrar los puntos de seguimiento a través de todos los fotogramas procesados hasta el momento. Son los *tracks* que requiere el proceso de triangulación.

7. Utilizar la triangulación multivista para calcular las posiciones iniciales en 3D correspondientes a las vistas introducidas hasta ese momento. Se realiza una reconstrucción cada vez que se añade una vista al viewSet, por tanto, se realizaran n-1 reconstrucciones, siendo n el número de fotogramas.

8. Utilizar *Bundle Adjustment* para perfeccionar las poses de la cámara y los puntos 3D. Guardar las poses de la cámara refinadas en el objeto viewSet. Se realiza este paso cada vez que se reconstruye mediante triangulación multivista para generar un modelo refinado y unas poses refinadas. Se vuelve al paso 2, hasta que se añade todas las vistas como fotogramas tenga el vídeo.

Todos los pasos de esta aproximación del algoritmo pueden observarse por medio de su reproducción visual en forma de diagrama de flujo.

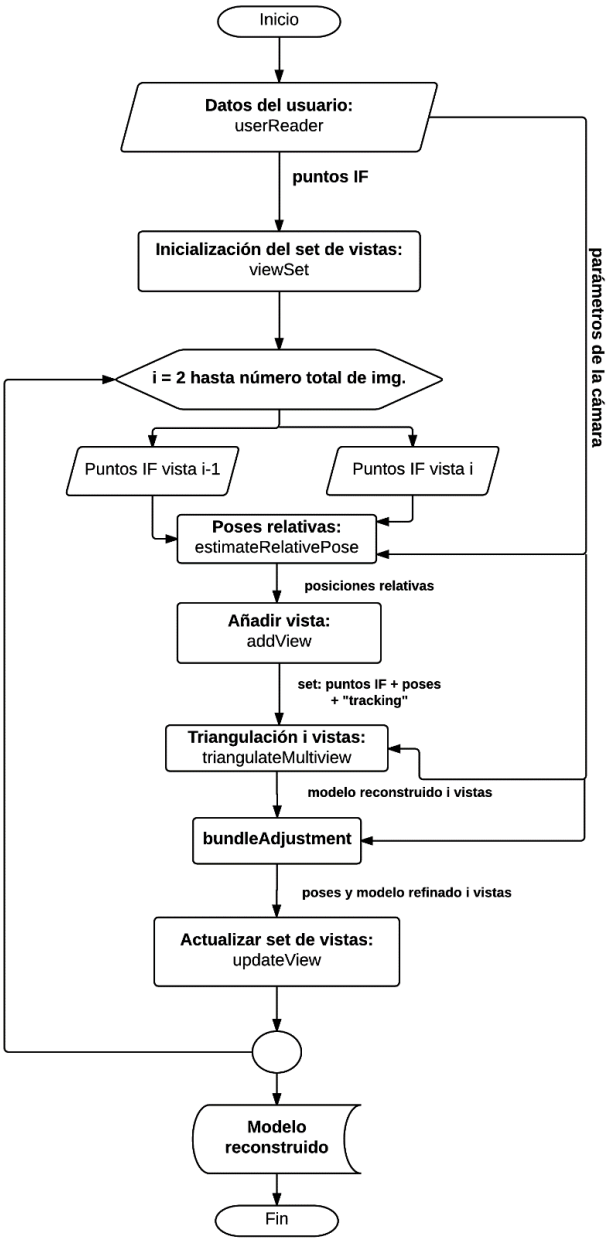


Figura 3.15. Diagrama de flujo del método implementado.

3.4.2. Elección de la técnica de estimación de *outliers*.

El método RANSAC, previo al cálculo de la Matriz Fundamental, realiza una discriminación previa de puntos para determinar aquellos que potencialmente pueden introducir un error importante en la obtención de los puntos de *IntraFace*.

En los algoritmos que implementen el cálculo de las poses relativas de la cámara a partir de la Matriz Fundamental junto con técnicas de exclusión de *outliers*, se siguen los siguientes pasos para calcular la Matriz Fundamental:

1. Se inicializa la matriz fundamental, F, una matriz de 3 por 3, formada por ceros.

2. Se ajusta el contador del bucle n, a cero, y el número de bucles de N, el número máximo de iteraciones especificado.

3. Bucle a través de los siguientes pasos, mientras n < N:
    a. Selección aleatoria de 8 pares de puntos de la imagen 1 y de la imagen 2.
    b. Uso de los 8 puntos seleccionados para calcular una matriz fundamental, f, utilizando el algoritmo de 8 puntos normalizado.
    c. Se calcula el ajuste de f para todos los puntos en la imagen 1 y la imagen 2.
    d. Si el ajuste de f es mejor que F, se reemplaza F por f. Si el método escogido es RANSAC o MSAC, se actualiza el valor de N.
    e. n = n + 1.

MATLAB permite emplear los métodos LMedS, RANSAC, MSAC y LTS en el proceso de selección de *inliers*. El método escogido es RANSAC, debido a su robustez y la posibilidad de influir en su resultado no determinista por medio de la introducción de parámetros, tal como el intervalo de confianza y el umbral de distancia para encontrar los *outliers*.

La elección se ha basado en un estudio realizado por S. Choi, T. Kim, W. Yu, donde se evalúa la precisión y el rendimiento de los distintos algoritmos o variantes dentro de la familia RANSAC [15].

Las siguientes tablas presentan un conjunto grande de técnicas de exclusión de *outliers* donde aparecen la mayoría de los métodos que implementa MATLAB. Se ponen a prueba para evaluar su robustez o precisión, y el rendimiento computacional de cada uno de ellos. Las pruebas se han realizado por medio de un conjunto de datos que siguen la función de una recta. Estos son los *inliers*. Estos datos se contaminan por medio de ruido o *outliers* que siguen una distribución gaussiana sin sesgo.

Los valores que se muestran se corresponden al error cuadrático normalizado de *inliers* o NSE, como definen Choi y Kim. Este error se calcula de manera similar al MSE, calculando la distancia del modelo calculado respecto a un *ground truth*. Este valor será cercano a 1 cuanto mayor sea la similitud con el *ground truth*.

Inlier Ratio	0.300	0.400	0.500	0.600	0.700	0.800	0.900
LMedS	**27.320**	**6.461**	1.321	1.356	1.408	1.379	1.509
RANSAC	1.305	1.323	1.326	1.330	1.390	1.415	1.483
MSAC	1.229	1.266	1.284	1.337	1.373	1.415	1.535
MLESAC	1.248	1.269	1.289	1.316	1.358	1.410	1.446
LO-RANSAC	1.245	1.229	1.229	1.221	1.229	1.253	1.255
R-RANSAC.T	1.317	1.323	1.304	1.341	1.394	1.401	1.475
R-RANSAC.S	**15.210**	1.848	1.131	1.229	1.291	1.370	1.389
FH' MAPSAC	**7.708**	**2.036**	1.647	1.484	1.463	1.490	1.526
AMLESAC	**2.051**	1.477	1.452	1.529	1.517	1.526	1.575
GASAC	**28.640**	**7.370**	1.108	1.077	1.100	1.120	1.147
pbM-estimator	1.023	1.034	1.209	1.255	1.286	1.291	1.355
uMLESAC	**5.246**	1.382	1.402	1.383	1.398	1.433	1.489
RANSAC*	**47.010**	**13.920**	**3.031**	1.688	1.386	1.255	1.187
MLESAC*	**50.110**	**10.240**	**2.839**	1.694	1.352	1.235	1.145

Tabla 3.1. NSE para cada método en función de la proporción de *inliers*.

Inlier Ratio	0.300	0.400	0.500	0.600	0.700	0.800	0.900
LMedS	0.226	0.094	0.047	0.025	0.015	0.009	0.005
RANSAC	0.220	0.091	0.046	0.025	0.015	0.009	0.005
MSAC	0.216	0.089	0.045	0.024	0.014	0.009	0.005
MLESAC	0.257	0.107	0.053	0.029	0.017	0.011	0.006
LO-RANSAC	0.232	0.098	0.051	0.029	0.018	0.012	0.008
R-RANSAC.T	0.220	0.091	0.045	0.011	0.008	0.006	0.005
R-RANSAC.S	0.038	0.052	0.055	0.012	0.011	0.017	0.029
FH'MAPSAC	0.046	0.052	0.038	0.024	0.015	0.010	0.006
AMLESAC	**1.336**	0.520	0.248	0.129	0.072	0.044	0.025
GASAC	**1.366**	0.569	0.286	0.157	0.093	0.060	0.036
pbM-estimator	**3.390**	**1.438**	0.731	0.406	0.236	0.155	0.091
uMLESAC	**1.072**	0.139	0.068	0.036	0.021	0.012	0.007
RANSAC*	0.014	0.014	0.015	0.014	0.014	0.014	0.015
MLESAC*	0.017	0.017	0.017	0.017	0.017	0.017	0.017

Tabla 3.2. Tiempo de cálculo para cada método en función de la proporción de *inliers*.

De las tablas anteriores, se llega a la conclusión en la que existe un cierto umbral de proporción de *inliers*, 0.6, donde todos los métodos se comportan de manera óptima. Valores inferiores a éste tienen un impacto inmediato en algunos métodos. El NSE crece de manera exponencial en los dos últimos métodos, RANSAC* y MLESAC*, en los cuales se ha sintonizado de manera manual sus respectivos parámetros, de manera fija. En los métodos que nos concierne, RANSAC, MSAC y LMedS, vemos que todos ellos convergen de manera similar, como se refleja en el tiempo de cálculo. LMedS es más sensible y siempre se aconseja para su correcto funcionamiento que se emplee si se conoce que más del 50% de los datos son *inliers*, como se muestra en la Tabla 3.1.

3.4.3. Limitaciones en la primera aproximación del algoritmo.

Realizando nuestras pruebas de manera experimental aplicadas a la base de datos sintética para la primera aproximación del método (algoritmo adaptado a puntos de *IntraFace* a partir del propuesto por MATLAB), se observa que el hecho de emplear un método estadístico y, por tanto, no determinista, en una tarea tan decisiva como es la estimación de las poses de la cámara, tiene un gran impacto en los resultados obtenidos en las reconstrucciones comparándolos con el *ground truth.* Vemos que son altamente variables dependiendo del método escogido en el cálculo de la Matriz Fundamental. Además, entre ejecuciones secuenciales manteniendo la misma configuración obtenemos mucha disparidad en la reconstrucción, por lo que no podemos considerar esta aproximación realizable de manera práctica. Se obtienen tanto resultados correctos como descartables.

Por ejemplo, para el usuario 5, que presenta la mayor deriva en la detección respecto al *ground truth*, relacionado con la menor proporción de *inliers*, este fenómeno es más acusado, como se presenta a continuación.

Del estudio de los apartados anteriores, se escoge el método RANSAC con una configuración de 0.5 para el intervalo de confianza y 0.1 para el umbral de búsqueda de *outliers*, suficiente para encontrar un compromiso muy adecuado entre tiempos de convergencia y rendimiento del mismo. Esta configuración se establece como estándar en todos los métodos del proyecto que requieran del cálculo de la Matriz Fundamental. Entre fotogramas, la cabeza presenta una diferencia de posición de 1º en *yaw*. En la siguiente tabla, se indica el número de la prueba ejecutada en la primera columna (**Nº TEST**), el MSE que se obtiene en la reconstrucción (**REC.-GT3D)** y los valores de los puntos de mayor y menor MSE (**MAX** y **MIN**, respectivamente).

Nº TEST	REC. - GT3D	MIN	MAX
1	4,44	0,38	14,51
2	17,30	4,31	33,03
3	5,23	0,88	16,60
4	12,50	3,33	24,16
5	10,94	2,90	22,02
6	33,17	8,00	60,94
7	24,86	6,97	42,31
8	8,83	2,59	19,28
9	31,11	8,79	61,34
10	22,72	5,20	41,40
MEDIA	17,11	4,34	33,56

Tabla 3.3. Variabilidad de la primera aproximación del algoritmo en el usuario 5.

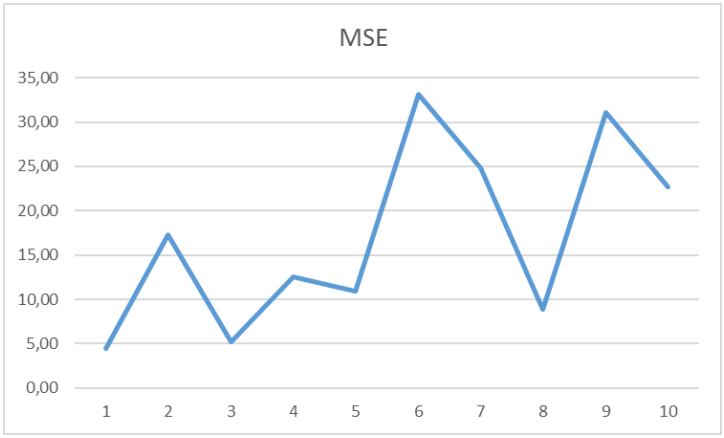


Figura 3.16. Gráfica que muestra la alta variabilidad de la primera aproximación del algoritmo en el usuario 5.

No obstante, se realizado el mismo experimento para el usuario 4, que tiene la menor distancia media con el *ground truth*. La realidad es que se reproduce un comportamiento muy similar en cuanto a la variabilidad entre ejecuciones, como en el usuario 5, ratificando que el procedimiento no funciona de manera correcta en ningún caso.

Nº TEST	REC. - GT3D	MIN	MAX
1	27,68	11,63	48,02
2	14,74	5,13	27,82
3	10,59	3,53	22,14
4	15,79	4,27	29,21
5	19,53	9,45	34,80
6	6,37	2,06	14,59
7	18,38	5,97	33,94
8	28,61	7,70	55,10
9	16,42	5,83	30,78
10	5,39	0,87	12,00
MEDIA	16,35	5,64	30,84

Tabla 3.4. Variabilidad de la primera aproximación del algoritmo en el usuario 4.

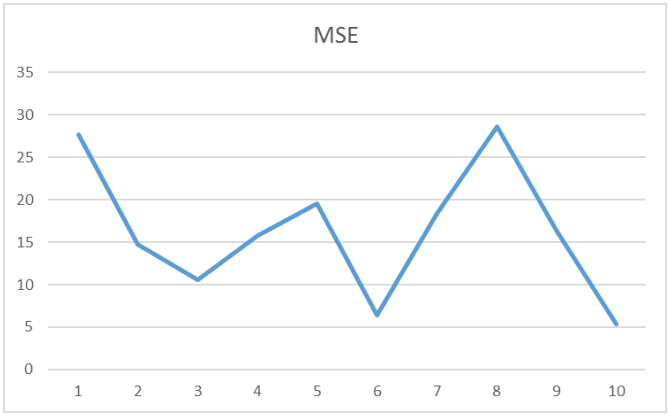


Figura 3.17. Gráfica que muestra la alta variabilidad de la primera aproximación del algoritmo en el usuario 4.

Debido a estos resultados, surge la necesidad de replantear de nuevo el método mediante otro enfoque que consiga mitigar en gran medida el impacto que tiene calcular las poses de la cámara mediante un método estadístico.

Uno de los principales motivos en la existencia de tal variabilidad entre resultados sucesivos (existe una probabilidad del 30% para obtener un MSE inferior a 10), es la selección de los 8 puntos iniciales en el consenso empleado por RANSAC. Si en este conjunto hay una cierta cantidad de *outliers* entre ellos, se intentará mejorar este conjunto de forma insatisfactoria, por tanto, el cálculo de la pose de la cámara será erróneo. Aunque se exija una fracción de validación elevada de 0.9 para la Matriz Fundamental, este pequeño error se acarrea en los siguientes cálculos en el refinamiento de la reconstrucción.

Esta pose errónea, junto a las anteriores, se refinan por medio de *Bundle Adjustment* cada vez que se añade una vista. *Bundle Adjustment* puede derivar en más error, ya que las primeras poses, si son erróneas, son las que tienen más peso en el algoritmo porque son las que más veces se emplean en esta función, dando lugar en un incremento de la deriva del error.

También se sabe de antemano que el cálculo de la Matriz Fundamental no es óptimo cuando los puntos se encuentran en una superficie casi plana, como en nuestro caso, un rostro. Bajo este supuesto, se debe hallar una forma para que los cambios en la localización de los puntos adopten mayores diferencias entre los fotogramas para minimizar esta indeterminación.

En el siguiente apartado, se plantea una nueva aproximación del algoritmo para solucionar estos problemas y conseguir mejores reconstrucciones, mitigando el factor no determinista sabiendo que siempre

se obtendrán resultados distintos entre ejecuciones con la misma configuración por la naturaleza del método RANSAC en la búsqueda de *inliers*.

3.4.4. Segunda aproximación del algoritmo.

Con objeto de solventar y mejorar el procedimiento anterior, se modifican ciertos aspectos del mismo para obtener reconstrucciones robustas y más deterministas frente a las ejecuciones del mismo. El problema principal es la alta variabilidad introducida por el cálculo de la Matriz Fundamental por las razones comentadas en el apartado anterior.

Primeramente, se realizará una única reconstrucción y, por tanto, una única llamada a la función de *Bundle Adjustment*. Este cambio mejorará este aspecto para estabilizar el comportamiento del algoritmo. Tiene como contrapartida en que nunca se podrá alcanzar el mejor resultado de la aproximación anterior, si se daba el caso que se escogían de manera correcta los *inliers* en los fotogramas porque se realizaban N-1 llamadas al algoritmo de refinado. Se obtendrán menores tiempos de ejecución por pasar a realizar una única reconstrucción en el algoritmo por medio del proceso de triangulación.

La segunda modificación está más orientada a su aplicación en la base de datos real, donde se presenta el fenómeno de las oclusiones y su impacto negativo en el cálculo de la Matriz Fundamental. Este cambio consiste en calcular las poses relativas de cámara respecto al primer fotograma, donde aseguramos que la cabeza adopta una posición completamente fija. A su vez, al ser frontal, *IntraFace* también tiene el menor error si lo comparamos con el *ground truth*.

El segundo planteamiento del algoritmo contempla estos cambios respecto al primer planteamiento. Por tanto, todos los pasos del algoritmo quedarían de la siguiente forma:

1. Entre un par formado por el primer y el fotograma actual, encontrar un conjunto de correspondencias de puntos. Se detectan los puntos de interés utilizando el algoritmo de *IntraFace* y se realiza el seguimiento de estos puntos mediante su algoritmo propio, combinando la detección en el fotograma actual con detecciones anteriores para mejorar la estimación.

2. Estimar la pose relativa de la vista actual respecto a la primera vista. Es la orientación de la cámara y la ubicación con respecto a la primera imagen. Se inicia un proceso que se encarga de calcular la Matriz Fundamental a partir de una serie de puntos correspondientes, localizados en dos fotogramas distintos. Con dicha matriz, se calcula las poses relativas entre ambas imágenes. Una función auxiliar puede permitirnos escoger el método de cálculo de la Matriz Fundamental, así como los parámetros asociados a cada uno, llamada *estimateRelativePose*. La familia de métodos RANSAC tienen un comportamiento estocástico debido a la naturaleza estadística del mismo. Se puede generar un bucle con una condición de salida tal que, la fracción de validación del cálculo de la Matriz Fundamental sea elevada para asegurar el correcto cálculo y aproximar en la máxima medida a un resultado determinista.

3. Almacenar los atributos de la vista actual: Pose de la cámara y de los puntos de imagen.

4. Almacenar las correspondencias *inlier* entre la primera vista y la vista actual. En este caso, todos los puntos son considerados *inlier* porque se ha realizado *tracking* y no *matching*.

5. Encontrar los puntos de seguimiento a través de todos los fotogramas. Son los *tracks* que requiere el proceso de triangulación.

6. Utilizar la triangulación multivista para calcular las posiciones iniciales en 3D correspondientes a los fotogramas.

7. Utilizar *Bundle Adjustment* para perfeccionar las poses de la cámara y los puntos 3D.

Todos los cambios de esta nueva aproximación del algoritmo pueden visualizarse por medio de esta reproducción en forma de diagrama de flujo.

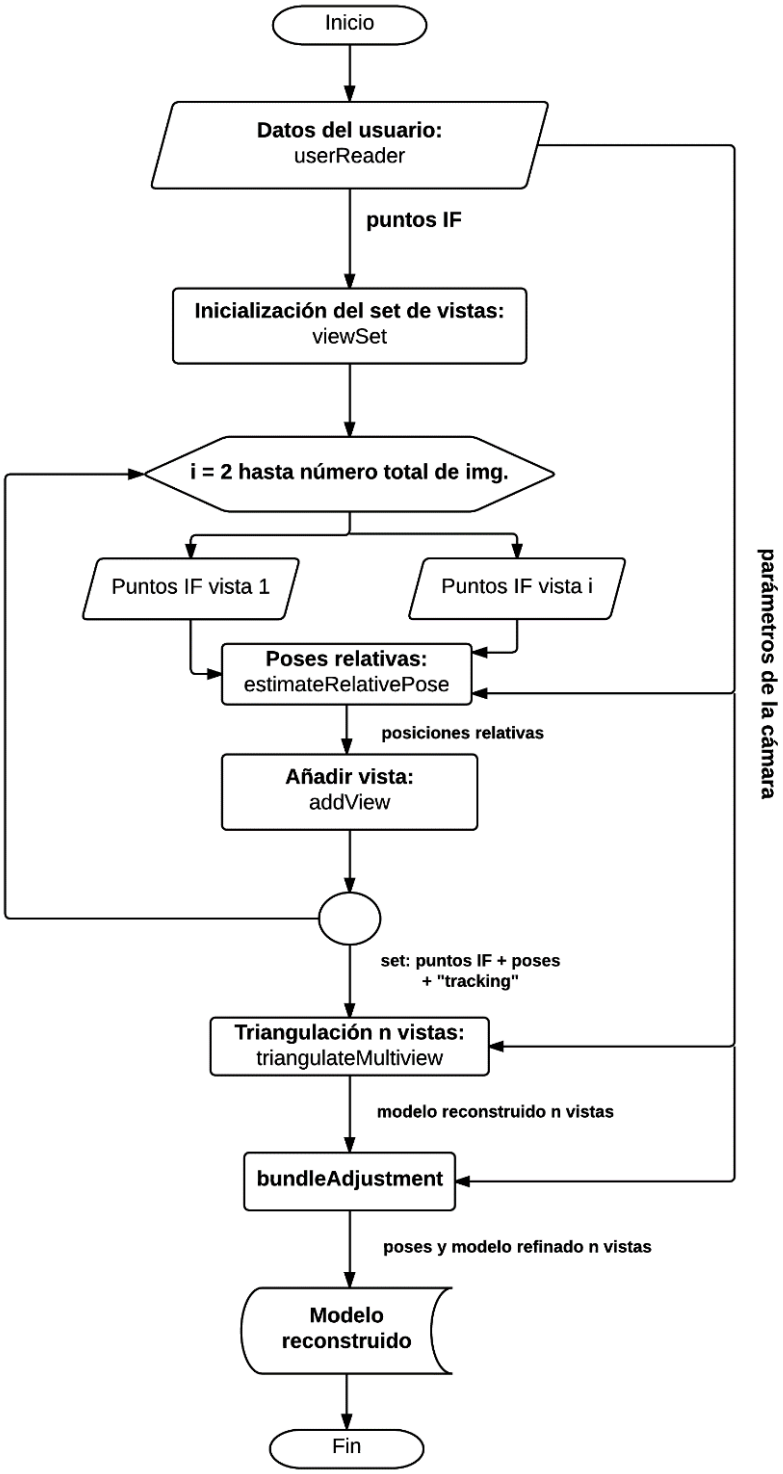


Figura 3.18. Diagrama de flujo del método implementado.

3.5. Valoración de los resultados obtenidos. Base de datos sintética.

3.5.1. MSE, error cuadrático medio.

En este apartado, se presentan los resultados obtenidos en el cálculo del MSE, empleando la segunda aproximación del método, ya que la primera aproximación no permite establecer unos resultados de manera consolidada. En este último caso, supondría realizar un gran número de ejecuciones del algoritmo para cada uno de los usuarios de la base de datos sintética y quedarnos con el mejor modelo de todos ellos, algo irrealizable en la práctica de manera trivial cuando no se dispone del *ground truth*.

Por otro lado, esta segunda versión arroja resultados muy similares entre ejecuciones sucesivas manteniendo la misma configuración del algoritmo: número de fotogramas, cambios de rotación en la cabeza entre fotogramas, parámetros del método de exclusión de *outliers*, etc.

Se demuestra para todos los métodos desarrollados que, incluir un mayor número de fotogramas, unido a la no presencia de saltos abruptos en la localización de los puntos, permite obtener mejores reconstrucciones, ya que la compensación del error de proyección que introduce *Bundle Adjustment* es más efectiva.

Puntos de *IntraFace* ideales o GT2D.

Como era previsible, los resultados en el uso de los puntos de *IntraFace* GT2D no suponen ninguna limitación ni ponen en duda la efectividad del algoritmo porque, en este caso, todos los puntos son considerados *inliers*, además de no presentar prácticamente ninguna variación por derivas en el *tracking* o *matching.* Cada punto es unívoco en cada fotograma.

Se muestran los resultados obtenidos en este escenario, donde se indica en la siguiente tabla cada usuario de la base sintética, el MSE en milímetros que hay entre la cabeza media y el modelo ideal (columna **MH – GT3D**), el MSE entre el modelo reconstruido con el modelo ideal (columna **REC. – GT3D**) y la distancia mínima (columna **MIN**) y máxima (columna **MAX**) entre dos puntos equivalentes entre el modelo refinado y el modelo 3D ideal de cada usuario.

USUARIO	MH - GT3D	REC. - GT3D	MIN	MAX
1	3,60	1,38	0,42	4,33
2	3,37	0,78	0,18	2,33
3	4,16	0,46	0,10	1,34
4	3,56	0,49	0,12	1,58
5	2,87	0,97	0,34	2,58
6	3,56	0,54	0,14	1,52
7	3,51	0,51	0,09	1,41
8	3,26	0,80	0,30	2,41
9	3,17	1,09	0,31	3,07
10	3,31	0,69	0,20	1,86
MEDIA	**3,44**	**0,77**	**0,22**	**2,24**

Tabla 3.5. Resultados para los 10 usuarios empleando los puntos característicos de *IntraFace* ideales o GT2D.

El algoritmo toma los 43 puntos de cada fotograma para calcular las poses de la cámara relativas, por tanto, los resultados son deterministas porque no tienen ningún efecto la aplicación del método RANSAC para excluir *outliers*.

Puntos de *IntraFace* promediados y sin promediar.

Los resultados del MSE de la reconstrucción usando de los puntos de *IntraFace* promediados y sin promediar son los que suscitan real interés en nuestra aplicación, ya han sido generados a partir de un algoritmo de localización de puntos faciales que incorpora su propio seguimiento de ellos en los sucesivos fotogramas. Estos puntos si presentan derivas, como se ha comprobado en el capítulo anterior. Es objeto de este algoritmo desarrollar un buen comportamiento ante esta deriva para estimar de manera correcta los *inliers* que existen entre fotogramas, para después calcular de manera precisa las poses de la cámara y manejar de esta manera la deriva existente en una aplicación real donde va a existir de forma intrínseca una componente de ruido en la estimación de estos puntos debido a muchos motivos: uniformidad del fotograma, resolución de la imagen, algoritmos de búsqueda de correspondencia de puntos entre fotogramas (*matching* o *tracking),* etc.

En estos casos, sí que entra en funcionamiento el algoritmo de análisis de RANSAC para excluir *outliers*. Es una tarea crítica para establecer unas correctas poses relativas de la cámara calculadas a partir de los puntos que marca como *inliers* la salida del algoritmo.

Este proceso determina el funcionamiento general del proceso de reconstrucción. La siguiente figura muestra un ejemplo de discriminación de los 43 puntos que realiza RANSAC entre la primera imagen, que estipulamos como de referencia (vista frontal de la cabeza), respecto a esos 43 puntos observados en otro fotograma de un vídeo de un usuario de la base de datos real. Las líneas que unen puntos correspondientes entre las dos imágenes. Los puntos que son *outliers*, se denominan puntos putativos, porque no se consideran realmente puntos que pueden formar líneas epipolares. El funcionamiento óptimo de RANSAC nos elimina los puntos de este tipo.

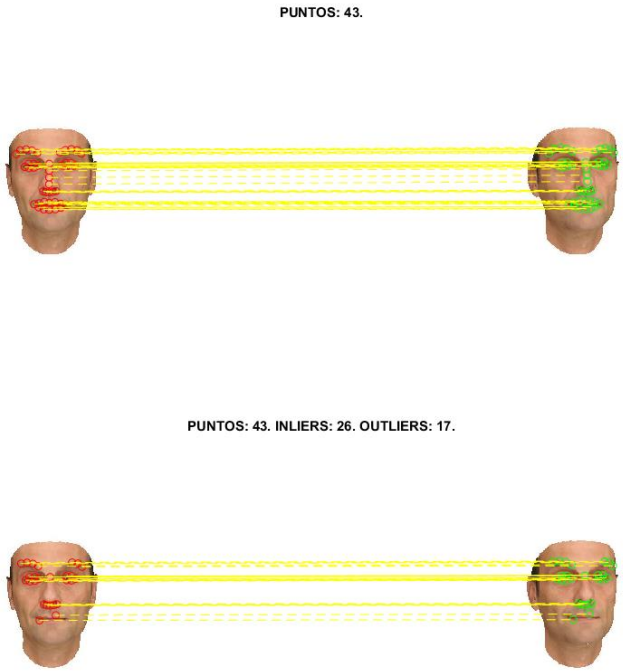


Figura 3.19. Ejemplo de funcionamiento de RANSAC para el usuario 8 de la base de datos sintética. Arriba, los 43 puntos de entrada al algoritmo de RANSAC. Abajo, discriminación de puntos establecida por este método a su salida.

De manera análoga a los resultados de las reconstrucciones empleando puntos GT2D, se presenta el MSE para todos los usuarios de esta base de datos, además del histograma normalizado de *inliers* acumulado a partir de todos ellos.

USUARIO	MH - GT3D	REC. - GT3D	MIN	MAX
1	3,60	3,82	0,32	15,87
2	3,37	4,53	1,23	16,86
3	4,16	1,45	0,35	4,00
4	3,56	6,45	1,46	27,04
5	2,87	2,60	0,18	6,94
6	3,56	2,93	0,53	6,97
7	3,51	5,14	1,53	12,15
8	3,26	4,93	1,06	10,73
9	3,17	2,88	0,47	7,39
10	3,31	2,26	0,57	6,98
MEDIA	**3,44**	**3,70**	**0,77**	**11,49**

Tabla 3.6. Resultados para los 10 usuarios empleando los puntos característicos de *IntraFace* promediados.

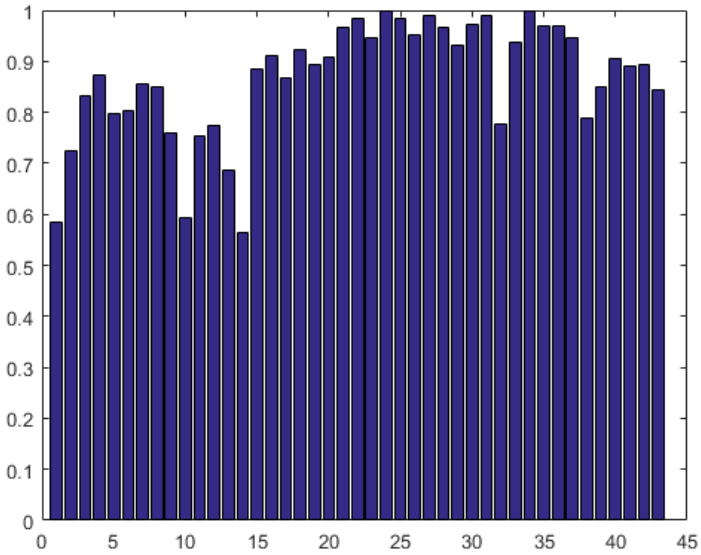


Figura 3.20. Histograma normalizado del uso de *inliers* acumulado para los 10 usuarios empleando los puntos característicos de *IntraFace* promediados.

USUARIO	MH - GT3D	REC. - GT3D	MIN	MAX
1	3,60	3,63	0,20	14,17
2	3,37	4,00	1,03	8,69
3	4,16	4,19	0,78	9,56
4	3,56	6,58	2,02	13,50
5	2,87	2,58	0,24	7,03
6	3,56	2,90	0,55	6,88
7	3,51	4,21	0,95	11,45
8	3,26	3,59	0,43	15,52
9	3,17	6,74	2,64	16,78
10	3,31	2,37	0,54	7,43
MEDIA	**3,44**	**4,08**	**0,94**	**11,10**

Tabla 3.7. Resultados para los 10 usuarios empleando los puntos característicos de *IntraFace* sin promediar.

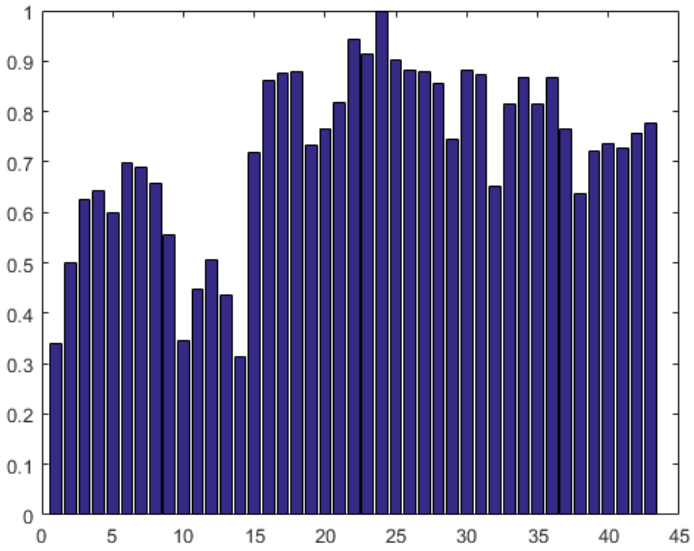


Figura 3.21. Histograma normalizado del uso de *inliers* acumulado para los 10 usuarios empleando los puntos característicos de *IntraFace* sin promediar.

De nuevo, no se encuentran diferencias apreciables en el MSE entre el uso de puntos promediados y sin promediar. Observamos que el algoritmo RANSAC hace una mayor exclusión de *outliers* cuando se emplean aquellos sin promediar y tiene un impacto incremental de 0.4 mm en el MSE respecto aquellas reconstrucciones donde se han tomado puntos promediados.

De los histogramas, vemos que ambos tienen un patrón similar en los puntos considerados *outliers*, lo que demuestra el buen comportamiento del método en diversos escenarios. Los puntos 1, 10 y 14 son los puntos de *IntraFace* que más se excluyen cuando los vídeos tienen movimiento de tipo *yaw*. Por otro lado, el rango de puntos de 16 a 29 son los que más se usan para calcular las poses relativas de la cámara. Estos puntos tienen la peculiaridad de que son los más visibles en todos los fotogramas.

3.5.2. Error en HPE.

Como parte a su vez del cálculo del error en la reconstrucción, se compara el modelo reconstruido con el *ground truth* del movimiento absoluto en rotación y traslación por medio de POSIT, para calcular su error en la estimación de la pose de la cabeza que se ha realizado por medio de la Matriz Fundamental.

El desconocimiento del factor de escala también es intrínseco a este método porque el cálculo que se realiza es a través de fotogramas, por tanto, es relativo. Poseer el GT3D soluciona este problema de escala, ya que podemos referenciar el tamaño a este, como se realiza en el cálculo del MSE a través de *Procrustes*.

En este apartado, el error elevado en traslación absoluto también es una constante, lo que lleva a referenciar tanto las poses del modelo como del *ground truth* al primer fotograma para hacer una estimación correcta.

Puntos de *IntraFace* ideales o GT2D.

	Absoluto			GT zeroed - HPE zeroed		
USUARIO	Tx	Ty	Tz	Tx	Ty	Tz
1	36,05	165,77	744,01	8,04	0,38	2,27
2	28,41	167,33	696,11	3,80	0,35	4,83
3	8,86	130,99	660,67	6,11	0,80	2,93
4	20,27	150,67	665,36	7,27	0,70	1,66
5	17,08	156,48	666,46	7,09	1,65	7,23
6	16,89	175,80	677,33	4,21	1,01	2,69
7	10,49	133,19	695,56	7,51	0,66	2,68
8	19,75	161,58	750,99	6,15	1,10	5,43
9	5,27	138,86	712,29	11,34	0,73	4,32
10	5,50	138,77	636,65	9,96	1,00	10,03
MEDIA	**16,86**	**151,94**	**690,54**	**7,15**	**0,84**	**4,41**
MEDIA T.	**286,45**			**4,13**		

Tabla 3.8. Error HPE en traslación. Puntos de *IntraFace* GT2D.

	Absoluto			GT - HPE zeroed			GT zeroed - HPE zeroed		
USUARIO	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)
1	1,03	1,15	0,94	1,10	1,09	0,93	0,10	0,98	0,02
2	3,03	1,09	5,35	1,81	0,85	5,28	0,21	0,80	0,33
3	1,01	0,71	1,30	0,78	0,67	1,37	0,10	0,64	0,04
4	0,94	0,69	1,05	0,99	0,65	1,06	0,05	0,63	0,14
5	3,61	1,07	1,11	2,19	0,90	0,99	0,26	0,80	0,16
6	1,59	0,72	1,33	1,50	0,70	1,30	0,08	0,63	0,05
7	1,80	0,67	1,85	1,72	0,71	1,84	0,07	0,61	0,03
8	1,85	1,01	1,63	1,07	0,73	1,59	0,19	0,67	0,13
9	2,19	1,47	1,08	1,00	2,46	1,30	0,14	1,03	0,18
10	3,20	1,60	1,73	1,59	1,50	1,93	0,37	0,76	0,14
MEDIA	**2,02**	**1,02**	**1,74**	**1,38**	**1,02**	**1,76**	**0,16**	**0,75**	**0,12**
MEDIA T.	**1,59**			**1,39**			**0,35**		

Tabla 3.9. Error HPE en rotación. Puntos de *IntraFace* GT2D.

Puntos de *IntraFace* promediados.

	Absoluto			GT zeroed - HPE zeroed		
USUARIO	Tx	Ty	Tz	Tx	Ty	Tz
1	36,04	165,76	743,88	8,03	0,38	2,27
2	31,19	166,67	699,38	1,32	1,40	24,23
3	2,05	121,81	639,96	2,04	5,46	38,65
4	12,21	149,09	675,73	1,86	2,34	20,82
5	17,04	156,47	664,80	7,04	1,64	7,22
6	16,81	175,68	674,97	4,19	1,01	2,66
7	10,50	133,18	696,00	7,52	0,66	2,68
8	19,75	161,57	751,09	6,15	1,10	5,43
9	5,28	138,88	711,27	11,34	0,74	4,35
10	5,47	138,85	633,94	9,92	0,97	10,04
MEDIA	**16,86**	**151,94**	**690,54**	**7,15**	**0,84**	**4,41**
MEDIA T.	**285,18**			**6,45**		

Tabla 3.10. Error HPE en traslación. Puntos de *IntraFace* promediados.

	Absoluto			GT - HPE zeroed			GT zeroed - HPE zeroed		
USUARIO	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)
1	1,50	1,86	0,96	1,79	2,07	0,94	0,90	1,86	0,54
2	2,81	0,23	6,97	2,94	0,54	6,99	3,20	0,15	0,19
3	9,07	0,62	9,56	9,05	0,87	9,56	1,68	0,59	0,19
4	3,18	0,62	6,82	3,28	0,84	6,85	3,32	0,21	0,12
5	1,54	0,58	0,88	2,39	0,71	0,96	0,30	0,42	0,21
6	1,83	0,96	1,11	1,44	1,07	1,11	0,35	0,96	0,38
7	1,57	2,59	1,96	1,52	2,67	2,10	0,43	2,65	0,25
8	1,45	2,09	1,71	1,19	2,09	1,62	0,44	2,10	0,18
9	1,18	2,48	1,51	1,68	2,42	1,23	0,66	2,63	0,16
10	1,84	1,50	2,06	2,48	1,62	1,90	0,74	1,72	0,63
MEDIA	**2,60**	**1,35**	**3,35**	**2,78**	**1,49**	**3,33**	**1,20**	**1,33**	**0,28**
MEDIA T.	**2,44**			**2,53**			**0,94**		

Tabla 3.11. Error HPE en rotación. Puntos de *IntraFace* promediados.

3.6. Valoración de los resultados obtenidos. Base de datos real.

3.6.1. MSE, error cuadrático medio.

Aplicando el algoritmo diseñado a la base de datos real, obtenemos unas reconstrucciones que son, en esencia, aquellas que se obtienen en una aplicación real donde se parte de un conjunto de vídeos grabados a través de una cámara con su respectiva distorsión y no idealidad. Respecto a las imágenes, la calidad de éstas es determinante en el procesado y obtención de puntos mediante el algoritmo de *IntraFace* o por medio de detectores de otro tipo de puntos. También, como se ha visto en el capítulo anterior, los puntos presentan oclusiones cuando el sujeto real no mantiene constante su expresión facial. Por tanto, este escenario más realista presenta un mayor reto para el algoritmo que la base de datos sintética.

Puntos de *IntraFace* promediados.

Las limitaciones para obtener resultados para el MSE usando puntos de *IntraFace* GT2D se mantienen al disponer los mismos modelos de referencia GT3D, como ocurría en el capítulo anterior. Realizando el diezmado y reordenación oportuna del modelo GT3D de 54 puntos con el modelo reconstruido de 43 puntos, se obtiene de igual manera un MSE aproximado en el caso que empleemos puntos de *IntraFace* promediados.

USUARIO	MH - GT3D	REC. - GT3D	MIN	MAX
1	6,05	6,21	0,29	13,01
2	7,08	6,93	1,90	17,21
3	6,94	5,82	1,51	14,75
4	6,69	7,26	1,30	18,48
5	6,20	4,47	1,57	7,67
6	6,25	5,50	0,61	11,63
7	6,24	4,89	0,44	12,80
8	7,35	5,85	0,69	12,19
9	6,88	5,05	0,86	8,52
10	8,45	6,60	0,41	15,51
MEDIA	**6,81**	**5,86**	**0,96**	**13,18**

Tabla 3.12. Resultados para los 10 usuarios empleando los puntos característicos de *IntraFace* promediados.

Como en el caso de la base de datos sintética para puntos promediados, se adjunta la demostración en la que el algoritmo toma las correspondencias de los 43 puntos entre un par de fotogramas, hace una discriminación de puntos, y calcula las poses relativas de la cámara con los puntos que RANSAC ha considerado *inliers*.

La siguiente figura muestra el ejemplo de funcionamiento de RANSAC aplicado a un usuario de la base de datos real.

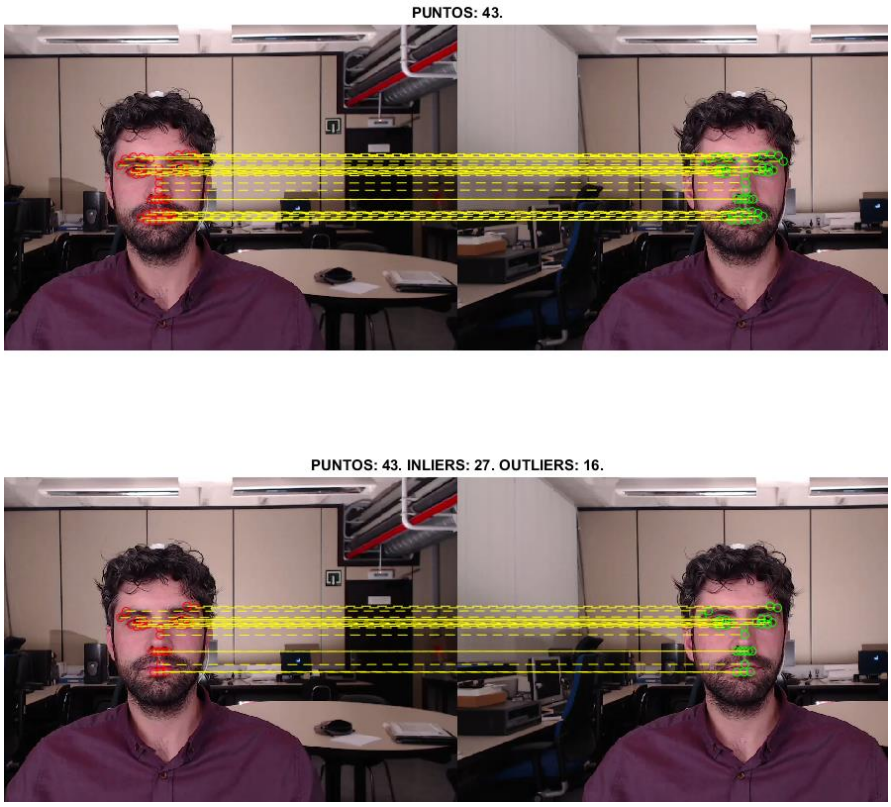


Figura 3.22. Ejemplo de funcionamiento de RANSAC para el usuario 2 de la base de datos real. Arriba, los 43 puntos de entrada al algoritmo de RANSAC. Abajo, discriminación de puntos establecida por este método a su salida.

La siguiente tabla se corresponde a un histograma que se ha normalizado respecto al valor más alto de concurrencia, el punto que más veces se ha considerado *inlier*.

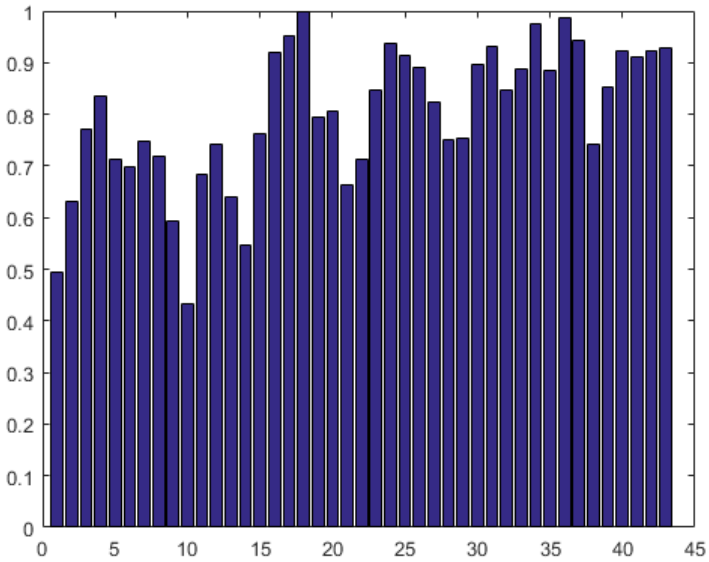


Figura 3.23. Histograma normalizado del uso de *inliers* acumulado para los 10 usuarios empleando los puntos característicos de *IntraFace* promediados.

3.6.2. Error en HPE.

Puntos de *IntraFace* promediados.

	Absoluto			GT zeroed - HPE zeroed		
USUARIO	Tx	Ty	Tz	Tx	Ty	Tz
1	37,14	165,32	745,65	8,11	0,38	2,28
2	27,86	166,94	697,64	3,85	0,35	4,85
3	8,99	130,44	661,81	6,17	0,81	2,92
4	19,31	150,03	665,42	7,31	0,69	1,64
5	17,96	155,99	667,41	7,15	1,67	7,27
6	17,57	175,59	679,46	4,23	1,01	2,70
7	10,81	132,81	697,96	7,58	0,67	2,70
8	18,91	160,98	751,91	6,14	1,10	5,41
9	5,61	138,38	714,06	11,35	0,72	4,32
10	5,24	138,33	638,23	9,98	1,02	10,03
MEDIA	**16,94**	**151,48**	**691,96**	**7,19**	**0,84**	**4,41**
MEDIA T.	**286,79**			**4,15**		

Tabla 3.13. Error HPE en traslación. Puntos de *IntraFace* promediados.

	Absoluto			GT - HPE zeroed			GT zeroed - HPE zeroed		
USUARIO	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)
1	3,06	1,78	1,26	2,32	0,90	1,20	1,48	0,66	1,03
2	1,10	3,18	5,02	1,02	3,11	4,79	1,33	3,14	0,62
3	2,76	2,01	2,33	1,61	1,98	2,13	1,21	1,99	1,09
4	1,83	4,25	1,63	1,29	4,16	1,57	0,41	4,17	0,81
5	1,48	1,25	1,25	2,98	0,77	0,77	0,77	0,79	0,84
6	3,05	2,22	1,07	3,15	1,55	1,04	2,16	1,63	1,00
7	2,03	3,41	1,57	2,04	2,29	1,56	0,93	1,73	1,58
8	2,21	1,92	2,01	1,48	1,95	1,82	1,85	1,87	0,55
9	1,92	4,81	1,93	1,33	3,66	1,91	1,09	2,51	0,89
10	2,95	3,99	2,70	1,61	2,79	2,05	1,32	2,10	0,61
MEDIA	**2,24**	**2,88**	**2,08**	**1,88**	**2,31**	**1,89**	**1,26**	**2,06**	**0,90**
MEDIA T.	**2,40**			**2,03**			**1,40**		

Tabla 3.14. Error HPE en rotación. Puntos de *IntraFace* promediados.

4. Reconstrucción tridimensional de un modelo de cabeza mediante detección y descripción de puntos SURF, BRISK. Estimación de las poses de la cámara por medio de la Matriz Fundamental.

4.1. Introducción.

En los anteriores capítulos, se partía de la base en la que ya se introducían a los algoritmos una serie de puntos detectados de antemano mediante un algoritmo externo de detección de puntos faciales, en nuestro caso, *IntraFace*, para cada fotograma en particular y para cualquier vídeo mediante técnicas propias de seguimiento de esos puntos. Además, nos limitábamos a un número determinado de puntos, 43, condicionados por la deriva generada en la detección de estos puntos por las limitaciones de este algoritmo de detección.

Por tanto, anteriormente nos hemos centrado en las fases posteriores del proceso de la reconstrucción mediante SfM para un modelo de cabeza, como es el cálculo de las poses relativas de la cámara y el proceso de triangulación y refinado del modelo obtenido. En este capítulo, se abordará la primera fase del proceso de reconstrucción por medio de SfM: extracción de puntos y seguimiento o detección de correspondencias en las sucesivas imágenes.

Tener libertad para escoger el tipo de puntos a detectar mediante numerosos algoritmos de detección de puntos presenta una gran ventaja, ya que permite contrastar distintos y variados métodos que incorpora la plataforma MATLAB y ajustarnos a uno para una aplicación específica.

Para la reconstrucción tridimensional, lo más común es hacer uso de descriptores que sean invariantes a escala y rotación, es decir, generar una localización de puntos robusta frente a cambios en las poses de la cámara entre sucesivas imágenes, así como la posibilidad de hallar sus respectivas correspondencias. Los descriptores que cumplen dichas características son SURF, FREAK y BRISK. MATLAB implementa dos de ellos: SURF y BRISK.

También se puede escoger entre dos métodos para encontrar correspondencias de puntos:

- ***Matching***: consiste en localizar de manera independiente una serie de puntos en dos imágenes distintas empleando el mismo detector de características locales. Tras estas detecciones, se describen las regiones que contienen estos puntos con características locales mediante descriptores. Los descriptores se comparan entre ambas imágenes. Si existe correlación entre descriptores, los puntos contenidos en dicha descripción se consideran correspondencias.

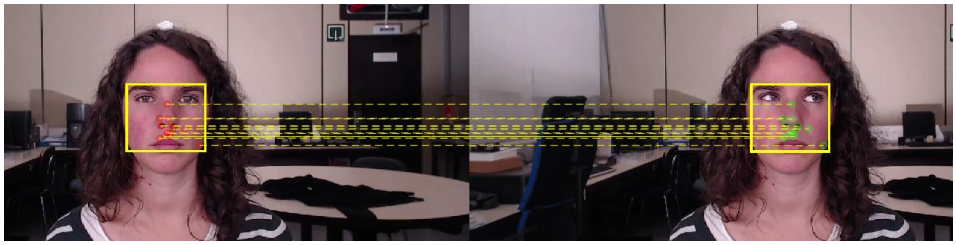


Figura 4.1. Ejemplo de *matching* de puntos SURF para el usuario 07 de la base de datos real.

- ***Tracking***: consiste en realizar una única detección de puntos en una imagen e inicializar un algoritmo de *tracking* mediante éstos, para que se realice una estimación de su desplazamiento en la siguiente imagen. Esta estimación del desplazamiento se corresponde al seguimiento de los puntos. El método más empleado para realizar *tracking* de puntos es el algoritmo de Lucas-Kanade-Tomasi (KLT), por ser eficiente y rápido en el proceso de registro de imágenes que suele ser, generalmente, costoso.

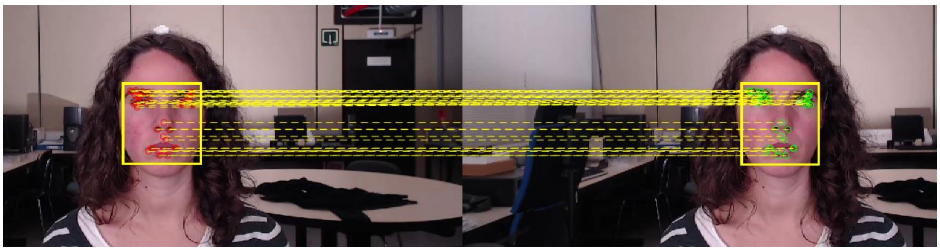


Figura 4.2. Ejemplo de *tracking* de puntos SURF para el usuario 07 de la base de datos real.

Una vez hecha esta detección de puntos para todos los fotogramas de un determinado de vídeo, los siguientes pasos se realizarán de manera convencional respecto a lo implementado hasta el momento, como es el cálculo de las poses de la cámara mediante la Matriz Fundamental, método que establecimos como general para cualquier tipo y cantidad de puntos, si existen correspondencia entre ellos.

4.2. Diseño del algoritmo de reconstrucción.

4.2.1. Primera aproximación del algoritmo: *Matching*.

La primera aproximación del algoritmo basado en la detección de puntos de los diversos fotogramas consiste en realizar una extensión de la segunda aproximación del algoritmo planteado en el capítulo anterior, modificando el primer paso de éste para implementar nuevas técnicas de detección y seguimiento de puntos distinta a la realizada por *IntraFace*. Consiste en localizar una serie de puntos arbitrarios mediante detección y descripción de características locales. Los puntos detectados serán de tipo SURF o BRISK, supeditados a los detectores válidos para nuestra aplicación que incorpora MATLAB. Tienen las propiedades necesarias para nuestra aplicación en concreto, sobre todo por permitir buscar correspondencias de imágenes, además de ser robustos ante cambios en la rotación, escala o intensidad.

Para señalizar de manera correcta al detector la zona de la imagen donde debe realizar la detección de los puntos, con el fin de desestimar aquellas que carecen de interés o pueden ser fuentes de ruido para el proceso de reconstrucción (zonas que no pertenecen a la cabeza, o el fondo de la imagen), se define una región de interés o *Region Of Interest* (por sus siglas en inglés, ROI) para acotar la zona en la detección de puntos.

El proceso de reconstrucción se inicia definiendo una ROI en cada fotograma. Es una región rectangular definida, en este caso, por la distancia entre los puntos más distantes de *IntraFace* en anchura y en altura. En dicha región se realiza una detección de puntos de interés con una configuración determinada. Cada detector permite variar una serie de parámetros propios. Modificarlos tiene una incidencia directa en la calidad y número de puntos detectados, por lo que es recomendable ajustarlos de manera óptima para encontrar un compromiso adecuado en el número, ya que se debe trabajar con una cantidad mínima de 8 puntos para poder calcular la Matriz Fundamental. Un gran número de puntos detectados puede tener un impacto negativo en el proceso de *matching*, ya que se incrementa la posibilidad de tomar correspondencias erróneas si los puntos no tienen la calidad suficiente.

En el caso del detector SURF, pueden manejarse tres parámetros:

- **Umbral métrico:** selecciona un umbral de detección de los puntos con características más fuertes. Disminuir este valor devuelve un mayor número de puntos.
- **Número de octavas:** La detección se basa en la Diferencia de Gaussianos (DoG), que consiste en realizar un número determinado de escalados a la imagen, definidos como octavas. Cada octava tiene a su vez un número determinado de niveles que definen el tamaño de los filtros. Octavas superiores utilizan filtros más grandes y submuestrean en mayor medida las imágenes. Un mayor número de octavas dará como resultado la posibilidad de encontrar regiones de mayor tamaño. Debe establecerse un número adecuado de octavas dependiendo del tamaño de la imagen. Por ejemplo, en una imagen de 50x50 no se debe emplear más de 2 octavas.
- **Número de escalas por nivel:** parámetro que controla el número de filtros utilizados por octava. Al menos tres niveles son necesarios para analizar los datos en una sola octava.

Las regiones que contienen estos puntos se describen por medio de un descriptor, que puede ser SURF o BRISK, dependiendo de la detección, aunque existe la posibilidad de escoger el método de descripción de manera manual. La descripción se representa en forma de vector con longitud de 64 o 128. A mayor longitud, mayor precisión, pero disminuye la velocidad en la descripción.

En el siguiente fotograma se realiza este mismo procedimiento y se realiza un *matching* de los puntos para hallar correspondencias. En el *matching*, se puede escoger un valor de umbral de relación máximo para evitar hallar correspondencias erróneas o ambiguas. Modificar este valor permite representar un mayor o menor número de correspondencias.

Los siguientes pasos, relacionados con la obtención de las poses relativas de la cámara y triangulación de los puntos se realizan de manera idéntica a lo definido en la segunda aproximación del segundo capítulo, con la única diferencia que las poses relativas de la cámara se calculan entre sucesivas imágenes y no respecto al primer fotograma. Se encuentra un mayor número de correspondencias ante cambios de rotación pequeños.

Se puede describir el algoritmo basado en la detección de puntos definidos por características locales y búsqueda de correspondencias de éstos entre las sucesivas imágenes por medio de *matching*, en esta serie de pasos:

1. Definir una región de interés (ROI) en cada una de las imágenes para realizar la detección de los puntos.

2. Para cada par de imágenes consecutivas, encontrar un conjunto de correspondencias de puntos. Se detectan los puntos de interés utilizando funciones tales como *detectSURFFeatures*, para puntos SURF, o *detectBRISKFeatures*, para puntos BRISK.

3. Extracción de los descriptores utilizando *extractFeatures*. Entre los métodos de extracción de descriptores válidos se encuentra SURF y BRISK. BRISK tiene la principal de ventaja de ser un método con poco coste computacional, ya que la descripción es binaria. La descripción se realiza por medio de un vector. Se escoge la mayor longitud en el vector de descripción para aumentar la precisión, 128.

4. Se encuentran las correspondencias utilizando *matchFeatures* entre los descriptores de dos imágenes consecutivas.

5. Estimar la pose relativa de la vista actual. Es la orientación de la cámara y la ubicación con respecto a la imagen anterior. Se inicia un proceso que se encarga de calcular la Matriz Fundamental a partir de una serie de puntos correspondientes, localizados en dos fotogramas distintos. Con dicha matriz, se calcula las poses relativas entre ambas imágenes. Una función auxiliar puede permitirnos escoger el método de cálculo de la Matriz Fundamental, así como los parámetros asociados a cada uno, llamada *estimateRelativePoseEstimators*. La familia de métodos RANSAC tienen un comportamiento estocástico debido a la naturaleza estadística del mismo. Se puede generar un bucle con una condición de salida tal que, la fracción de validación del cálculo de la Matriz Fundamental sea elevada para asegurar el correcto cálculo y aproximar en la máxima medida a un resultado determinista.

6. Transformar la pose relativa de la vista actual al sistema de coordenadas de la primera imagen de la secuencia. La Matriz Fundamental se calcula entre imágenes sucesivas. Se transforman las poses obtenidas de acuerdo al sistema de referencia del primer fotograma.

   orientación actual = orientación anterior * orientación relativa actual
   localización actual = localización anterior + localización relativa actual * orientación anterior

7. Almacenar las correspondencias *inlier* entre la primera vista y la vista actual. Se regresa al segundo paso hasta completar el proceso en todos los fotogramas.

8. Encontrar los puntos de seguimiento a través de todos los fotogramas. Son los *tracks* que requiere el proceso de triangulación.

9. Utilizar la triangulación multivista para calcular las posiciones iniciales en 3D correspondientes a los fotogramas.

10. Utilizar *Bundle Adjustment* para perfeccionar las poses de la cámara y los puntos 3D.

Este algoritmo describe de manera integral todos los pasos de un procedimiento de reconstrucción mediante SfM. Todos los recursos necesarios para cada función son calculados en el mismo algoritmo, razón por la cual se convierte en un procedimiento interesante en aplicaciones prácticas sin necesidad de introducir elementos al sistema que han sido previamente calculados. No obstante, no se puede obviar el proceso de calibración de la cámara, pues es una etapa fundamental de SfM y necesaria para un gran número de pasos del procedimiento de los cuales hay una dependencia de los parámetros intrínsecos de la cámara. En resumen, cualquier vídeo que se introduzca al sistema, únicamente debe cumplir que se haya realizado con una cámara calibrada aportando al sistema sus respectivos parámetros intrínsecos.

Todos los cambios de esta nueva aproximación del algoritmo pueden visualizarse por medio de esta reproducción en forma de diagrama de flujo.

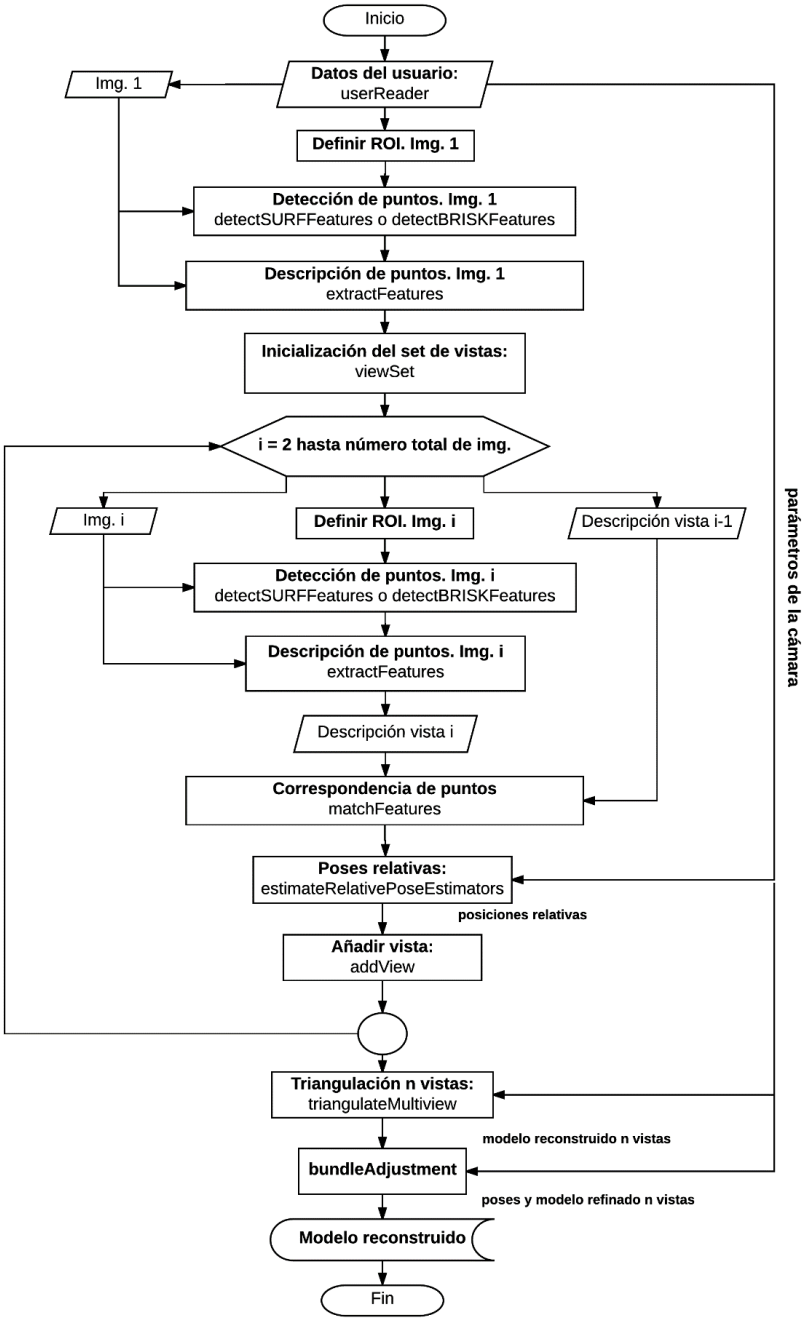


Figura 4.3. Diagrama de flujo de la primera aproximación del algoritmo.

4.2.2. Limitaciones en la primera aproximación del algoritmo.

El algoritmo parece funcionar adecuadamente si la detección y *matching* de puntos es lo suficientemente robusta. La limitación fundamental de este algoritmo está en la cantidad de puntos detectados en una imagen, pues no se trata de un número fijo. Con una misma configuración, es posible encontrar discrepancias en el número de puntos entre dos usuarios. El algoritmo de detección analiza ciertas características locales como pueden ser esquinas o *blobs*, lo que hace impredecible el número de puntos detectados, ya que depende de las superficies analizadas. Por tanto, no es posible elegir una configuración única para la detección de puntos mediante SURF para todos los usuarios.

La base de datos sintética presenta problemas en encontrar un número suficiente de correspondencias para realizar adecuadamente el proceso de obtención de las poses relativas, cuando no se alcanza un número suficientemente elevado de puntos para calcular de manera correcta la Matriz Fundamental. Esto se debe a la naturaleza de las cabezas de BFM, ya que no se tiene una suficiente resolución en las texturas, presentando superficies demasiado uniformes. Este fenómeno limita el resultado en la reconstrucción del algoritmo, ya que existe mucha dependencia del vídeo que introduzcamos al sistema.

En la base de datos real, las imágenes presentan más detalles, por tanto, es posible detectar un mayor número de puntos, aunque debe emplearse tanto en la base de datos real como sintética la configuración más elástica del detector, es decir, aquella que maximice el número de puntos localizado. Se puede realizar haciendo pequeño el umbral métrico y aumentando también el valor umbral de relación máxima para que el *matching* no sea demasiado exigente.

Debido a que se realizan múltiples detecciones a lo largo del algoritmo, no es posible trabajar con los mismos puntos de manera continuada, añadiendo además que el número de puntos no se mantiene constante.

A través de una representación de *boxplot*, se presentan los resultados para la base de datos sintética donde se refleja la variabilidad en la cantidad de puntos tras el *matching* entre dos fotogramas. Solo ha sido posible tomar resultados hasta el usuario 7, ya que en el usuario 8 el algoritmo no es capaz de avanzar porque no puede calcular la Matriz Fundamental en un cierto fotograma.

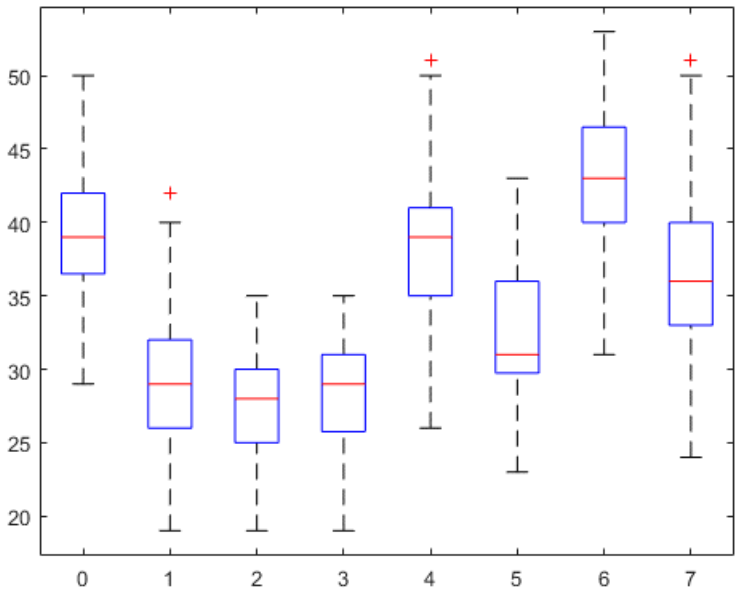


Figura 4.4. *Boxplot* para los 7 primeros usuarios de la base de datos sintética donde se representa la variabilidad en la cantidad de puntos tras el *matching* entre fotogramas.

Así como ocurría en la primera aproximación del capítulo anterior, existe también mucha variabilidad entre reconstrucciones con una configuración fija y tras sucesivas ejecuciones, incluso haciendo un *matching* estricto, es decir, exigiendo correspondencias únicas para evitar correspondencias ambiguas y una relación máxima de 0.1 para escoger las correspondencias más exactas. Estos motivos nos llevan a desestimar este algoritmo para aplicaciones prácticas.

Puesto que el algoritmo basado en *matching* presenta dichas limitaciones, la siguiente aproximación del algoritmo basado en detección de características locales es realizar un seguimiento de las características locales, de forma que se puede definir un conjunto inicial de puntos fijo y, además, si los cambios en la posición de la cabeza son relativamente pequeños respecto a fotogramas sucesivos, el *tracker* es capaz de mantener los mismos y la misma cantidad de puntos a través de fotogramas.

4.2.3. Segunda aproximación del algoritmo: *Tracking*.

De manera similar a lo que ocurría en el capítulo anterior para la primera aproximación del algoritmo, surge la necesidad de definir un nuevo enfoque en el algoritmo para obtener resultados robustos e invariables frente a las ejecuciones del mismo. Puesto que anteriormente no era posible mitigar el impacto estocástico del método RANSAC en la elección de *inliers*, debido a la alta variabilidad de correspondencias en los puntos entre distintos fotogramas, la siguiente opción es realizar una nueva implementación en la localización de puntos homólogos entre fotogramas realizando un seguimiento de ellos mediante un algoritmo de *tracking*. MATLAB incorpora una función basada en el algoritmo de Kanade-Lucas-Tomasi (KLT) de seguimiento de características. Sirve tanto para estabilización de vídeo, estimación del movimiento de la cámara y seguimiento de objetos [16].

Esta técnica está basada en el seguimiento de los puntos característicos, mediante las ecuaciones desarrolladas por Lucas-Kanade para el cálculo del flujo óptico, que a su vez implementan el método iterativo de Newton-Raphson para una búsqueda de ascenso por gradiente. Posteriormente se calcula para cada punto su vector de desplazamiento respecto al tiempo. No obstante, y como se puede deducir de un método iterativo de ascenso por gradiente, esta aproximación tan sólo es válida si el desplazamiento es lo suficientemente pequeño como para que el gradiente no cambie de dirección. Por ello y para desplazamientos mayores, se debe aplicar dicho procedimiento de forma piramidal [17].

Una vez realizada la puesta en correspondencia de los puntos característicos entre las dos imágenes consecutivas, es necesario realizar una corrección de perspectiva sobre alguna de ellas. Para ello se obtiene la homografía que permite la proyección de las coordenadas espaciales desde una imagen a otra, y viceversa. Dicha matriz de transformación homográfica actúa como matriz de proyección entre los espacios de coordenadas de ambas imágenes, permitiendo así no sólo hacer la corrección de perspectiva, sino que, en caso de no hacerse un registro de imagen, también puede ser necesario su uso dentro de los algoritmos de seguimiento.

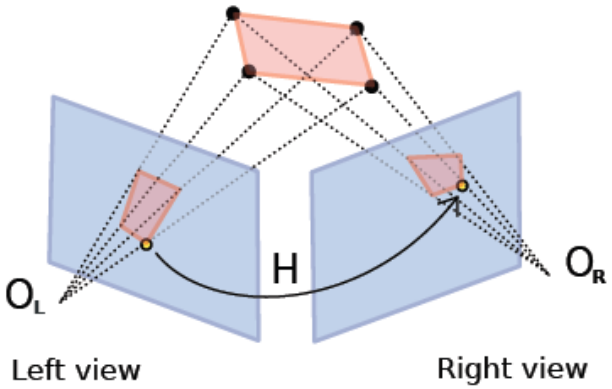


Figura 4.5. Relación homográfica entre dos vistas.

La forma de obtener dicha matriz de proyección es mediante el cálculo de la matriz de transformación *H*, comúnmente denominada homografía. Esta matriz permite proyectar los puntos pertenecientes al plano de una vista concreta, sobre el plano de otra vista diferente. De forma análoga a los métodos ya descritos, el algoritmo genera una deriva en el *tracking* que se deriva en la generación de *outliers* en el conjunto de datos. Una forma de resolver la exclusión de *outliers* para tomar puntos estables y robustos es a través de la técnica RANSAC.

Emplear *tracking* permite realizar una detección única de puntos en el primer fotograma. Posteriormente, se inicializa el algoritmo KLT con dichos puntos y se realiza un seguimiento a través de fotogramas. En cada paso, se almacena la nueva posición de los puntos y se actualiza el *tracker*. Por tanto, el *tracking* tiene unas ventajas notables respecto al *matching*:

- **Flexibilidad:** mediante *tracking*, es posible definir un conjunto inicial de puntos totalmente controlados por el usuario, tanto en cantidad, como en disposición espacial. No es necesario estar supeditados a detectores válidos para encontrar correspondencia de puntos, como SURF o BRISK. Es posible emplear cualquier detector de puntos característicos disponible.
- **Continuidad**: la cantidad y correspondencia de puntos entre fotogramas se mantiene constante. La limitación del *matching* radica en la variación en la elección de los puntos a través de los fotogramas, así como la cantidad.

Es posible definir una configuración del *tracker* basado en KLT a través de los siguientes parámetros con el fin de optimizar su comportamiento respecto a la aplicación en cuestión:

- **Número de niveles piramidales:** la implementación del seguimiento de puntos por medio del algoritmo KLT utiliza pirámides de imagen, donde en cada nivel se reduce la resolución en un factor dos en comparación con el nivel anterior. Seleccionar un nivel de pirámide mayor que 1 permite que el algoritmo pueda realizar un seguimiento de los puntos en los múltiples niveles de resolución, comenzando en el nivel más bajo. Aumentar el número de niveles piramidales permite que el algoritmo pueda manejar grandes desplazamientos de puntos entre fotogramas. Sin embargo, el coste computacional también aumenta.

  Cada nivel piramidal se forma por submuestreo del nivel anterior por un factor dos en anchura y altura. El *tracker* de puntos comienza el seguimiento de cada punto en el nivel más bajo de resolución, y continúa el seguimiento hasta que converge. Se propaga el resultado de ese nivel al siguiente nivel como estimación inicial de las ubicaciones de los puntos. De esta manera, el seguimiento se refina con cada nivel, hasta la imagen original. El uso de los niveles piramidales permite que el *tracker* pueda manejar grandes movimientos de píxeles, que pueden comprender distancias mayores que el tamaño de la vecindad.

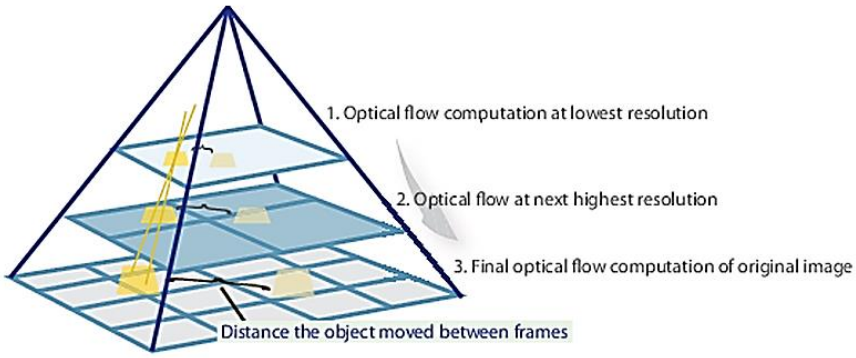


Figura 4.6. Ejemplo de niveles piramidales en una imagen.

- **Umbral de error bidireccional:** este valor es la distancia en píxeles desde la ubicación original de los puntos hasta la ubicación final después de realizar el seguimiento hacia atrás. Los puntos correspondientes se consideran no válidos cuando el error es mayor que el valor establecido para esta propiedad. Los valores recomendados son entre 0 y 3 píxeles. El error bidireccional es una forma efectiva de eliminar los puntos que no se han podido seguir con fiabilidad. Sin embargo, el error bidireccional requiere cálculo adicional [18].

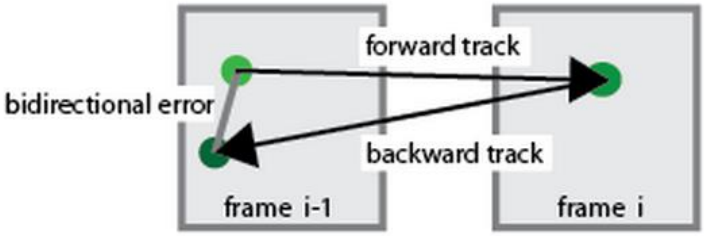


Figura 4.7. Ejemplo de cálculo del error bidireccional.

- **Tamaño de bloque:** se especifica un vector de dos elementos [altura, anchura] para representar la vecindad alrededor de cada punto que se está siguiendo. La altura y anchura deben ser enteros impares. La vecindad define el área para el cálculo de la matriz del gradiente espacial. El incremento en el tamaño del bloque aumenta el tiempo de cálculo.
- **Número máximo de iteraciones:** se puede definir un número máximo de iteraciones para el desarrollo de la búsqueda iterativa del algoritmo KLT en las nuevas localizaciones de los puntos. Generalmente, converge con 10 iteraciones.

De esta manera, se puede describir el algoritmo basado en la detección de puntos definidos por características locales y búsqueda de correspondencias de éstos entre las sucesivas imágenes por medio de *tracking*, en esta serie de pasos:

1. Definir una región de interés (ROI) en cada una de las imágenes para realizar la detección de los puntos iniciales.

2. Para el fotograma inicial, se realiza una detección de puntos o se define un conjunto de puntos por parte del usuario.

3. Se crea y se inicializa un *tracker* basado en el algoritmo KLT por medio del conjunto de puntos detectado o definido, junto al fotograma donde se ha realizado dicha definición.

4. Se realiza el seguimiento de tales en puntos en el fotograma siguiente y se comprueban las correspondencias de puntos, en el caso de que no haya sido posible seguir todos los puntos y se

hayan presentado algunas pérdidas, debido a oclusiones, cambios en la intensidad de luz o rotación o saltos abruptos en el movimiento. Para vídeos con mucha extensión temporal, es recomendable realizar una detección de puntos periódica.

5. Estimar la pose relativa de la vista actual. Es la orientación de la cámara y la ubicación con respecto a la imagen anterior. Se inicia un proceso que se encarga de calcular la Matriz Fundamental a partir de una serie de puntos correspondientes, localizados en dos fotogramas distintos. Con dicha matriz, se calcula las poses relativas entre ambas imágenes. Una función auxiliar puede permitirnos escoger el método de cálculo de la Matriz Fundamental, así como los parámetros asociados a cada uno, llamada *estimateRelativePoseEstimators*. La familia de métodos RANSAC tienen un comportamiento estocástico debido a la naturaleza estadística del mismo. Se puede generar un bucle con una condición de salida tal que, la fracción de validación del cálculo de la Matriz Fundamental sea elevada para asegurar el correcto cálculo y aproximar en la máxima medida a un resultado determinista.

6. Transformar la pose relativa de la vista actual al sistema de coordenadas de la primera imagen de la secuencia. La Matriz Fundamental se calcula entre imágenes sucesivas. Se transforman las poses obtenidas de acuerdo al sistema de referencia del primer fotograma.

   orientación actual = orientación anterior * orientación relativa actual
   localización actual = localización anterior + localización relativa actual * orientación anterior

7. Almacenar las correspondencias *inlier* entre la primera vista y la vista actual.

8. Actualizar el *tracker* con la posición actual del conjunto de puntos. Se regresa al cuarto paso hasta realizar el *tracking* en todos los fotogramas.

9. Encontrar los puntos de seguimiento a través de todos los fotogramas. Son los *tracks* que requiere el proceso de triangulación.

10. Utilizar la triangulación multivista para calcular las posiciones iniciales en 3D correspondientes a los fotogramas.

11. Utilizar *Bundle Adjustment* para perfeccionar las poses de la cámara y los puntos 3D.

Este algoritmo describe también de manera integral todos los pasos de un procedimiento de reconstrucción mediante SfM, como en la primera aproximación del algoritmo.

Empleando este procedimiento se consigue realizar todo el proceso de reconstrucción de una manera bastante robusta si el *tracker* se configura de manera adecuada. Para realizar las pruebas, se ha establecido una configuración computacionalmente exigente para asegurar el mejor funcionamiento del algoritmo KLT. La configuración escogida que asegura un buen comportamiento general en las dos bases de datos es de 8 niveles piramidales, tamaño de bloques de 21x21, sin cálculo de error bidireccional y un máximo de 30 iteraciones.

Todos los cambios de esta nueva aproximación del algoritmo pueden visualizarse por medio de esta reproducción en forma de diagrama de flujo.

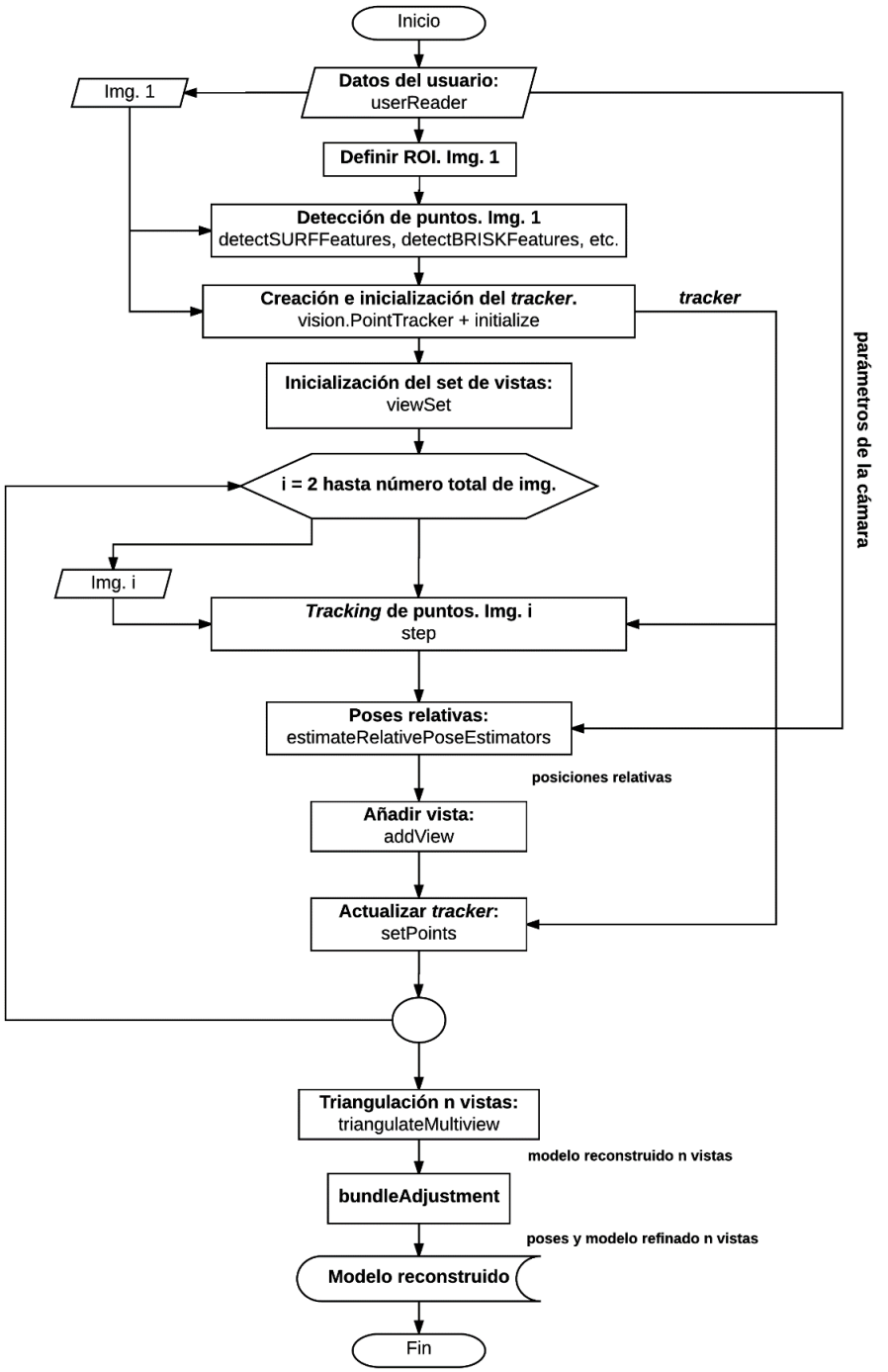


Figura 4.8. Diagrama de flujo de la segunda aproximación del algoritmo.

4.3. Valoración de los resultados obtenidos. Base de datos sintética.

4.3.1. MSE, error cuadrático medio.

En este apartado, se presentan los resultados obtenidos en el cálculo del MSE, empleando la segunda aproximación del método, es decir, a través de *tracking* de puntos, ya que la primera aproximación mediante *matching* no permite establecer unos resultados de manera consolidada. En este último caso, como ocurría en el capítulo anterior, supondría realizar un gran número de ejecuciones del algoritmo para cada uno de los usuarios de la base de datos sintética y quedarnos con el mejor modelo de todos ellos, algo irrealizable en la práctica de manera trivial cuando no se dispone del *ground truth*.

Esta segunda versión arroja resultados muy similares entre ejecuciones sucesivas manteniendo la misma configuración del algoritmo: configuración del *tracker*, número de fotogramas, cambios de rotación en la cabeza entre fotogramas, parámetros del método de exclusión de *outliers*, etc.

No obstante, para el correcto funcionamiento del *tracker*, para obtener buenos resultados se ha establecido un ángulo máximo de rotación en *yaw* de 20º, suficiente para mantener visibles todos los puntos y no estancar el seguimiento de algunos puntos cuando se sitúan justo en el borde de la cara.

A la hora de obtener el MSE mediante puntos SURF arbitrarios, nos encontramos con la problemática que no es posible calcularlo realizando POSIT con nuestro modelo GT3D definido mediante puntos de *IntraFace*. La solución planteada consiste en generar un conjunto de vistas con las poses de cámara calculadas normalmente mediante el algoritmo de *tracking* de puntos SURF y sustituir tales puntos por los puntos promediados de *IntraFace* en su respectiva vista. Mediante las poses de la cámara ya calculadas y los nuevos puntos, se reconstruye y se obtiene un modelo definido por puntos de *IntraFace*, suficiente para realizar el cálculo del MSE y el error en HPE. Por tanto, en dicha reconstrucción tiene gran peso las poses calculadas mediante el algoritmo propuesto.

Puntos de *IntraFace* promediados.

Se muestran los resultados obtenidos en este escenario, donde se indica en la siguiente tabla cada usuario de la base sintética, el MSE en milímetros que hay entre la cabeza media y el modelo ideal (columna **MH – GT3D**), el MSE entre el modelo reconstruido con el modelo ideal (columna **REC. – GT3D**) y la distancia mínima (columna **MIN**) y máxima (columna **MAX**) entre dos puntos equivalentes entre el modelo refinado y el modelo 3D ideal de cada usuario.

USUARIO	MH - GT3D	REC. - GT3D	MIN	MAX
1	3,60	2,31	0,80	7,67
2	3,37	2,89	0,45	9,42
3	4,16	1,08	0,26	3,26
4	3,56	5,22	1,90	15,58
5	2,87	7,36	0,37	19,76
6	3,56	2,59	0,48	8,08
7	3,51	8,25	2,06	19,44
8	3,26	2,83	1,09	7,71
9	3,17	3,65	0,25	10,98
10	3,31	6,74	0,63	20,49
MEDIA	**3,44**	**4,29**	**0,83**	**12,24**

Tabla 4.1. MSE para los 10 usuarios empleando los puntos característicos de *IntraFace* promediados.

4.3.2. Error en HPE.

Como parte a su vez del cálculo del error en la reconstrucción, se compara el modelo reconstruido con el *ground truth* del movimiento absoluto en rotación y traslación por medio de POSIT, para calcular su error en la estimación de la pose de la cabeza que se ha realizado por medio del algoritmo basado en el *tracking* de puntos arbitrarios.

El desconocimiento del factor de escala también es intrínseco a este método porque el cálculo que se realiza es a través de fotogramas, por tanto, es relativo. Poseer el GT3D soluciona este problema de escala, ya que podemos referenciar el tamaño a este, como se realiza en el cálculo del MSE a través de *Procrustes*.

En este apartado, el error elevado en traslación absoluto también es una constante, lo que lleva a referenciar tanto las poses del modelo como del *ground truth* al primer fotograma para hacer una estimación correcta.

Puntos de *IntraFace* promediados.

	Absoluto			GT zeroed - HPE zeroed		
USUARIO	Tx	Ty	Tz	Tx	Ty	Tz
1	34,62	165,03	715,12	6,99	0,40	2,09
2	26,06	165,86	651,90	3,15	0,43	4,33
3	7,29	133,47	610,76	4,23	0,68	2,34
4	18,81	150,82	616,30	5,12	0,63	1,81
5	15,88	156,49	625,12	4,55	1,39	6,45
6	15,26	173,08	641,26	3,88	1,03	2,53
7	8,88	135,79	645,61	5,20	0,64	3,03
8	19,57	161,77	716,29	5,50	0,89	5,66
9	4,95	139,39	690,08	10,71	0,73	4,32
10	5,12	139,91	588,49	9,08	0,44	9,42
MEDIA	**15,64**	**152,16**	**650,09**	**5,84**	**0,73**	**4,20**
MEDIA TOTAL	**272,63**			**3,59**		

Tabla 4.2. Error HPE en traslación. Puntos de *IntraFace* promediados.

	Absoluto			GT - HPE zeroed			GT zeroed - HPE zeroed		
USUARIO	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)
1	1,11	2,77	1,08	1,11	0,89	1,09	0,10	0,76	0,02
2	0,74	4,60	6,65	0,83	4,21	5,39	1,12	4,19	0,45
3	0,63	5,29	1,38	0,77	5,37	1,50	0,62	5,36	0,29
4	0,86	5,10	1,24	0,95	5,09	1,32	0,25	5,08	0,19
5	1,44	4,08	1,51	1,18	3,71	1,40	1,12	3,71	0,38
6	1,54	2,20	1,63	1,28	1,94	1,56	0,32	1,91	0,27
7	1,23	4,99	1,95	1,05	4,73	1,64	0,51	4,75	0,07
8	0,90	1,88	1,44	0,89	1,31	1,57	0,34	1,34	0,26
9	4,21	1,90	0,71	0,84	2,67	0,83	0,23	1,84	0,78
10	2,33	4,23	0,99	1,07	4,75	1,06	2,87	4,26	0,55
MEDIA	**1,50**	**3,70**	**1,86**	**1,00**	**3,47**	**1,74**	**0,75**	**3,32**	**0,33**
MEDIA TOTAL	**2,35**			**2,07**			**1,47**		

Tabla 4.3. Error HPE en rotación. Puntos de *IntraFace* promediados.

4.4. Valoración de los resultados obtenidos. Base de datos real.

4.4.1. MSE, error cuadrático medio.

Puntos de *IntraFace* promediados.

Las limitaciones para obtener resultados para el MSE usando puntos de *IntraFace* GT2D se mantienen al disponer los mismos modelos de referencia GT3D, como ocurría en el capítulo anterior. Realizando el diezmado y reordenación oportuna del modelo GT3D de 54 puntos con el modelo reconstruido de 43 puntos, se obtiene de igual manera un MSE aproximado en el caso que empleemos puntos de *IntraFace* promediados.

USUARIO	MH - GT3D	REC. - GT3D	MIN	MAX
1	6,05	6,20	0,20	13,02
2	7,08	6,33	2,18	16,34
3	6,94	6,35	0,83	15,83
4	6,69	5,29	0,42	11,46
5	6,20	8,48	0,95	25,69
6	6,25	6,49	0,99	17,36
7	6,24	5,03	0,69	12,99
8	7,35	8,69	1,74	22,44
9	6,88	4,71	1,68	8,99
10	8,45	7,94	0,47	22,39
MEDIA	**6,81**	**6,55**	**1,02**	**16,65**

Tabla 4.4. MSE para los 10 usuarios empleando los puntos característicos de *IntraFace* promediados.

4.4.2. Error en HPE.

Puntos de *IntraFace* promediados.

	Absoluto			GT zeroed - HPE zeroed		
USUARIO	Tx	Ty	Tz	Tx	Ty	Tz
1	42,80	161,05	717,73	7,19	0,39	2,11
2	20,68	162,57	668,46	3,39	0,35	4,65
3	9,55	128,62	634,24	4,91	0,75	2,74
4	12,22	146,63	635,14	5,53	0,62	1,48
5	25,88	150,49	624,69	4,87	1,50	6,89
6	23,23	168,78	643,66	3,99	1,09	2,64
7	14,37	131,37	666,05	5,65	0,69	2,70
8	11,99	156,66	718,78	5,53	0,90	5,56
9	10,62	135,60	690,82	10,77	0,69	4,42
10	5,03	135,43	610,91	9,35	0,77	9,41
MEDIA	**17,64**	**147,72**	**661,05**	**6,12**	**0,77**	**4,26**
MEDIA TOTAL	**275,47**			**3,72**		

Tabla 4.5. Error HPE en traslación. Puntos de *IntraFace* promediados.

	Absoluto			GT - HPE zeroed			GT zeroed - HPE zeroed		
USUARIO	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)	Roll (º)	Yaw (º)	Pitch (º)
1	2,36	0,65	1,24	2,34	1,01	1,25	1,46	0,66	1,01
2	1,54	1,71	4,66	1,03	1,80	4,78	1,12	1,89	0,69
3	2,72	2,74	2,06	1,47	2,79	1,96	1,14	2,81	0,96
4	1,44	1,07	1,94	1,34	0,98	1,95	0,58	1,01	1,08
5	0,80	5,44	1,07	0,86	5,50	1,11	1,69	5,54	0,24
6	2,88	2,46	1,01	2,65	2,42	0,99	1,93	2,48	0,61
7	1,44	2,25	1,33	1,36	1,98	1,22	0,95	1,52	0,91
8	0,95	3,21	1,73	1,21	3,23	1,73	1,72	3,29	0,33
9	2,47	3,26	1,59	1,37	3,59	1,60	0,81	1,38	0,74
10	2,26	3,41	1,77	0,95	3,18	1,74	2,77	2,59	0,36
MEDIA	**1,89**	**2,62**	**1,84**	**1,46**	**2,65**	**1,83**	**1,42**	**2,32**	**0,69**
MEDIA TOTAL	**2,12**			**1,98**			**1,48**		

Tabla 4.6. Error HPE en rotación. Puntos de *IntraFace* promediados.

4.5. Reconstrucción densa.

4.5.1. Introducción.

Para concluir este capítulo, se procederá a reconstruir los modelos de ambas bases de datos con un número elevado de puntos gracias a la libertad que nos proporciona el algoritmo en la elección del conjunto inicial de puntos. Para ello, se tomará el conjunto de vistas obtenido por cualquier algoritmo desarrollado y se realizará de nuevo un *tracking* con un número elevado de puntos y se obtendrán unas localizaciones de éstos en cada fotograma, para realizar de este modo una reconstrucción densa con las poses de la cámara ya calculadas previamente en el conjunto de vistas inicial. Es un procedimiento similar al empleado en el cálculo del MSE, mediante puntos de *IntraFace* para los algoritmos de este capítulo.

El desafío fundamental del procedimiento que se va a presentar a continuación es la elección del conjunto inicial de puntos. Tiene que ser lo suficiente elevada para tener una resolución adecuada para calcular y definir, posteriormente, superficies a partir de estos puntos. Además, la superficie debe ser continua, de forma que la distancia entre los puntos tiene que ser, mayormente, constante.

4.5.2. Procedimiento.

El procedimiento es totalmente aplicable a cualquier algoritmo descrito en este proyecto, pues se realiza posteriormente a obtener un conjunto de vistas, lo que da más libertad para escoger las poses de la cámara mejor calculadas por alguna de ellos. El método necesita a su entrada la introducción de los fotogramas, parámetros de la cámara para realizar el proceso de triangulación y *Bundle Adjustment*, de manera que se obtiene la reconstrucción densa y, por último, el conjunto de vistas que contiene las poses de la cámara para cada fotograma.

Para obtener una reconstrucción densa de un modelo de cabeza, se define una rutina que se puede resumir en los siguientes pasos:

1. El primer paso consiste en definir una región de interés (ROI), rectángulo descrito por cuatro valores: x de la esquina superior izquierda, y de la esquina superior izquierda, anchura y altura. Acota el fotograma a una región que abarca la parte principal de la cara, con el fin de desestimar la definición de puntos en zonas que carezcan de interés en la reconstrucción, como puede ser el fondo de la imagen.

2. A continuación, se define el conjunto inicial de puntos sobre el cual se realizará el *tracking* y serán aquellos que formen parte de la reconstrucción densa. Para ello, se crea una malla de puntos equidistantes dentro de la ROI, separados por un cierto valor de píxeles, que denominamos delta. La elección de este valor es fundamental en la reconstrucción, ya que debe ser lo suficientemente pequeño para para obtener una cantidad elevada de puntos, pero lo suficientemente grande para que el algoritmo sea escalable respecto las posibilidades del *tracker*. Se ha observado que, si escoge una distancia entre puntos de uno o dos píxeles, el *tracking* no converge y no es capaz de continuar. Un valor razonable que tenga en cuenta las consideraciones anteriores es de 4 píxeles.

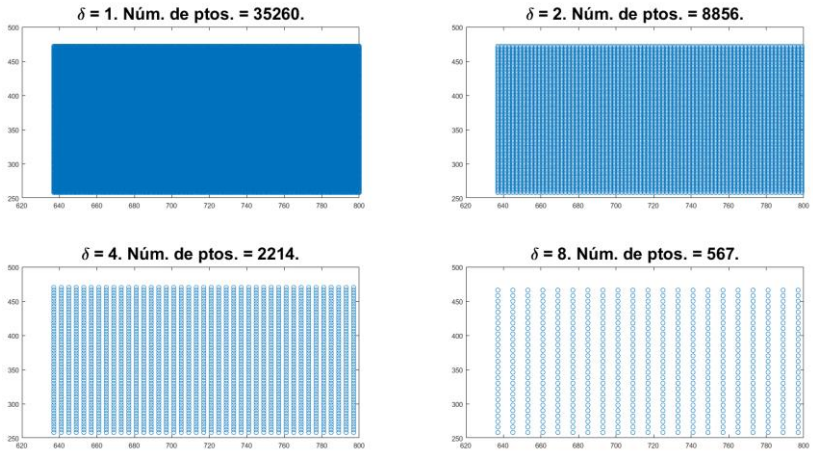


Figura 4.9. Comparación en la cantidad de puntos en función de delta para una ROI de [636, 257, 164, 215].

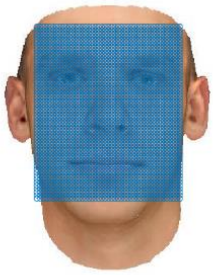


Figura 4.10. Malla de puntos equidistantes 4 píxeles definida en la ROI para el primer usuario de la base de datos sintética.

3. Se ajusta el *tracker* KLT con un umbral de error bidireccional de 1 píxel y un número de niveles piramidales de 4. Se inicializa con los puntos definidos en el primer fotograma.

4. Se realiza el seguimiento de tales puntos en el fotograma siguiente y se comprueban las correspondencias, en el caso de que no haya sido posible seguir todos los puntos y se hayan presentado algunas pérdidas debido a oclusiones, cambios en la intensidad de luz o rotación o saltos abruptos en el movimiento. Para vídeos con mucha extensión temporal, es recomendable realizar una detección de puntos periódica.

5. Actualizar el *tracker* con la posición actual del conjunto de puntos. Se regresa al cuarto paso hasta realizar el *tracking* en todos los fotogramas.

6. Encontrar los puntos de seguimiento a través de todos los fotogramas. Son los *tracks* que requiere el proceso de triangulación.

7. Utilizar la triangulación multivista para calcular las posiciones iniciales en 3D correspondientes a los fotogramas. Utilizar *Bundle Adjustment* para perfeccionar las poses de la cámara y los puntos 3D.

8. Es importante que la nube de puntos esté limpia de puntos espurios diseminados fuera de la ROI por alguna deriva del *tracking*. Este proceso de exclusión se realiza por medio de un simple análisis estadístico que consiste en determinar aquellos puntos que excedan un cierto valor de píxeles respecto a un cierto valor de percentil en cada eje.

Eje X: 100 píxeles respecto al percentil 60.
Eje Y: 100 píxeles respecto al percentil 30.
Eje Z: 100 píxeles respecto al percentil 60.

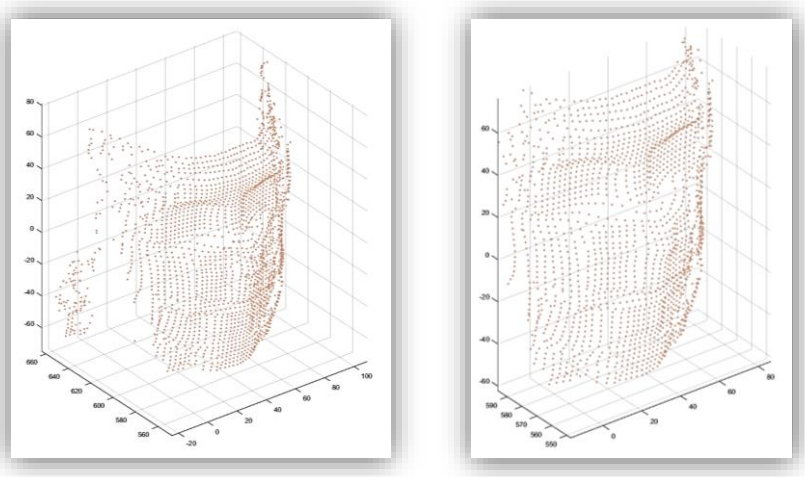


Figura 4.11. A la izquierda, nube de puntos obtenida tras la reconstrucción. A la derecha nube de puntos tras realizar la exclusión de puntos espurios.

9. Adicionalmente, una vez obtenida la reconstrucción densa de puntos, es posible realizar una representación volumétrica por medio de un elevado número de superficies definidas con un mínimo de tres puntos. Una función externa llamada *MyCrustOpen* realiza este procedimiento, obteniendo los vértices de los triángulos por medio del algoritmo de triangulación de *Delaunay* sobre la nube de puntos. En la representación, es posible parametrizar la incidencia de luz en las superficies, así como su procedencia, para dar un efecto más realista.

Si realizamos la reconstrucción densa para la base de datos real, se observa que las zonas más susceptibles a presentar cambios debido a gesticulación o cambios en la expresión de la cara, como en la zona de los ojos o la boca, se refleja en la representación como regiones con puntos con baja continuidad respecto a la vecindad, debido a estas oclusiones.

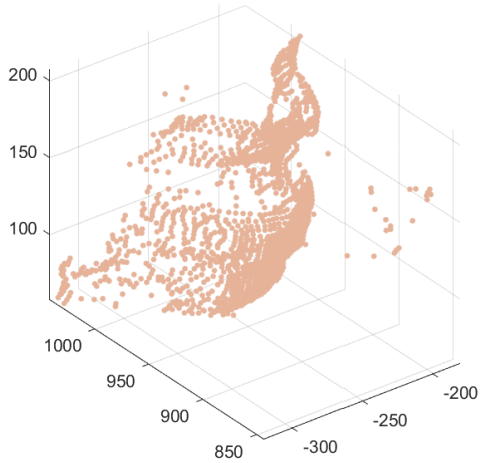


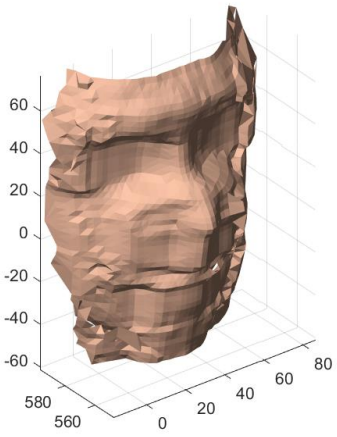
Figura 4.12. Reconstrucción densa del usuario 04 de la base de datos real. El modelo carece de continuidad en la zona de los ojos debido a las oclusiones.

A continuación, se muestran los resultados de las reconstrucciones densas de los usuarios de la base de datos sintética por medio de su representación volumétrica y el cálculo del MSE respecto al *ground truth*.

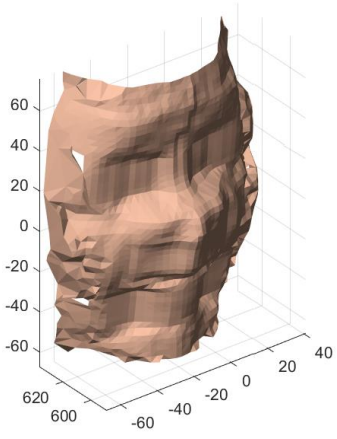
4.5.3. Resultados de la reconstrucción densa. Base de datos sintética.

Representación 3D

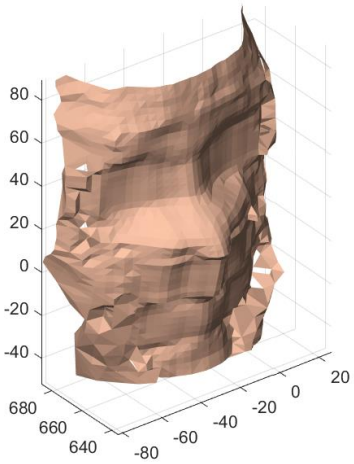
Usuario 01:



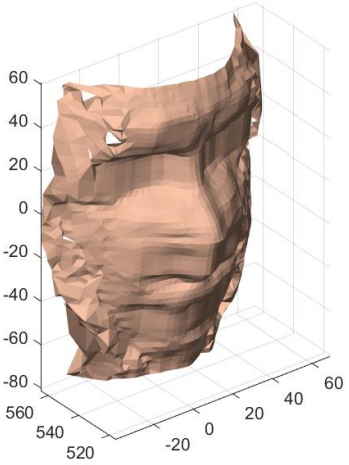
Usuario 04:



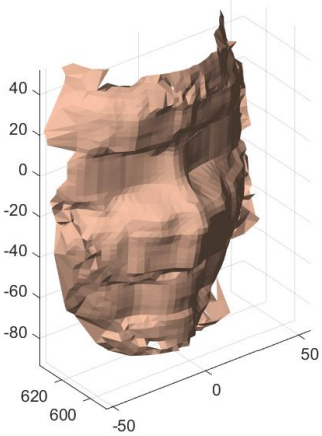
Usuario 02:



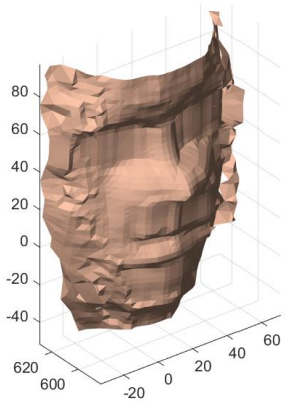
Usuario 05:



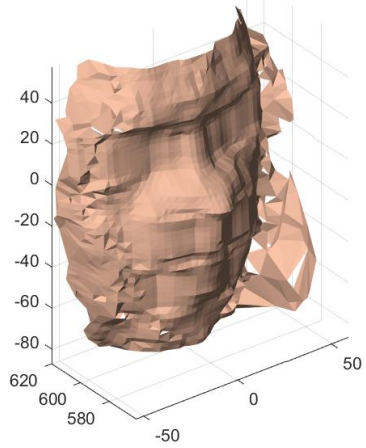
Usuario 03:



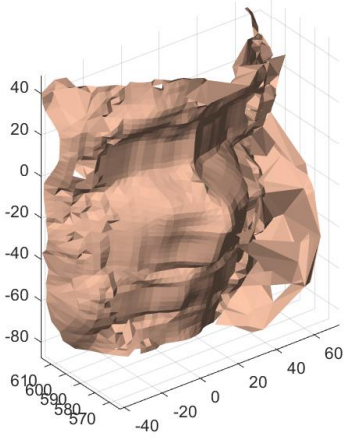
Usuario 06:



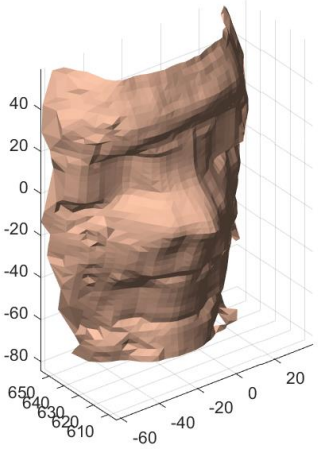
Usuario 07:



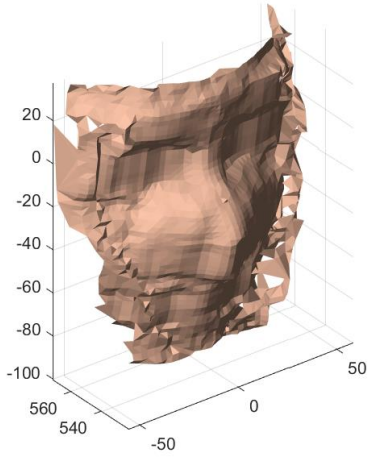
Usuario 09:



Usuario 08:



Usuario 10:



MSE

Además, puesto que tenemos el GT3D generado por BFM para cada usuario de la base de datos sintética con una gran resolución (53490 puntos), es posible obtener un valor de MSE entre las dos nubes densas, el GT3D y la nube obtenida en el proceso de reconstrucción, realizando un proceso de registro de ambas.

Este proceso consiste en normalizar primeramente la posición de ambos modelos a una localización espacial común, de forma que encuentren un solapamiento máximo entre ambas. Este procedimiento previo se denomina pre-registro. Es importante, para que el método de registro, en este caso el algoritmo ICP (*Iterative Closest Point*), no se detenga al encontrar un mínimo local que nos aleje del resultado correcto. ICP busca la transformación, mediante un proceso iterativo, que debe realizarle a una nube de puntos dada respecto a una de referencia para minimizar el error cuadrático entre ambas [19].

Una vez registradas las nubes de puntos, se calcula de manera habitual el MSE, a partir del promedio en todo el conjunto de distancias euclídeas de los puntos de la nube densa respecto a los puntos del GT3D más cercanos a éstos.

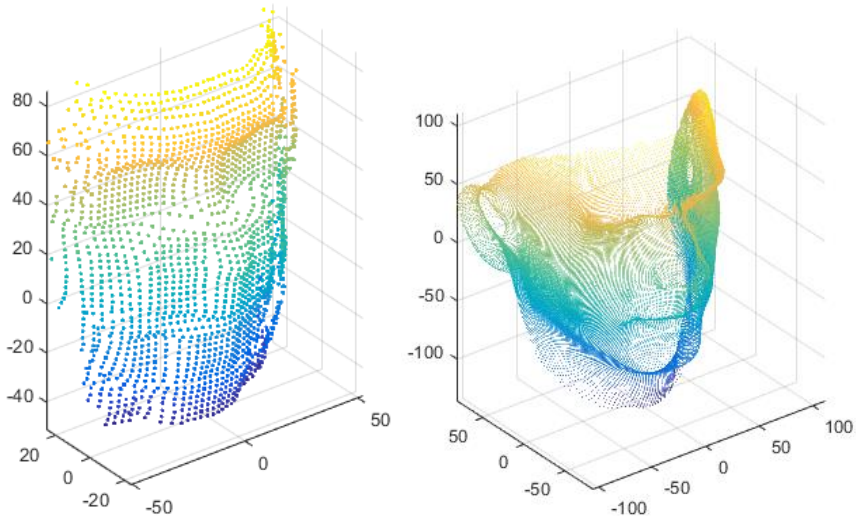


Figura 4.13. A la izquierda, pre-registro de la nube densa reconstruida. A la derecha, pre-registro del GT3D.

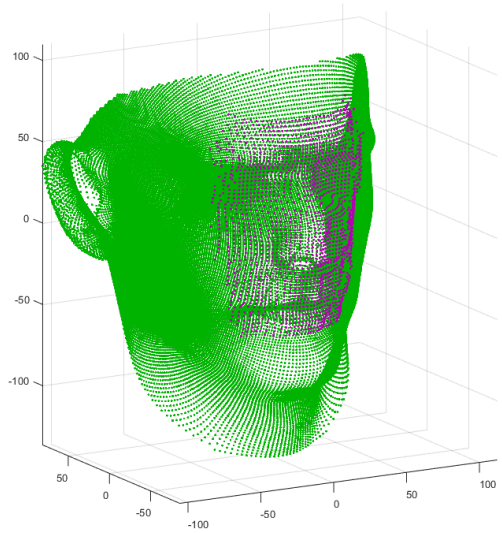


Figura 4.14. Registro de ambas nubes densas de puntos por medio de ICP.

USUARIO	MSE (mm)
1	4,01
2	2,71
3	3,15
4	2,97
5	2,97
6	2,73
7	4,22
8	3,27
9	4,89
10	2,85
MEDIA	**3,38**

Tabla 4.7. MSE de los 10 usuarios de la base de datos sintética.

5. Conclusiones.

5.1. Resumen de los resultados. Base de datos sintética.

En este capítulo se contrastarán y se resumirán los resultados obtenidos en todos los algoritmos desarrollados en este proyecto. Para la base de datos sintética, es posible reconstruir usando el GT2D, ya que el GT3D está definido para los puntos de *IntraFace*. Por tanto, es posible realizar POSIT y *Procrustes* para obtener el MSE y el error en HPE, respectivamente.

MSE.

Las siguientes tablas resumen los resultados obtenidos en los métodos desarrollados con resultados deterministas, y, por tanto, viables. Los valores se corresponden a los promedios para todos los usuarios de la base de datos sintética. Se define el método empleado en la detección de puntos y en el cálculo de las poses de la cámara. Verticalmente, están ordenados respecto al algoritmo representativo de cada capítulo. En la columna **REC. – GT3D**, se muestra el valor de MSE entre la reconstrucción final y el GT3D. Las columnas MIN y MAX se corresponden al MSE de los puntos de *IntraFace* en la reconstrucción con menor y mayor MSE, respectivamente.

MÉTODO		MSE (mm)		
DETECCIÓN PTOS.	CÁLCULO DE POSES DE LA CÁMARA	REC. - GT3D	MIN	MAX
IntraFace	POSIT + cabeza media	0,09	0,02	0,29
IntraFace	Matriz Fundamental	0,77	0,22	2,24

Tabla 5.1. MSE promedio para todos los usuarios de la base de datos sintética. GT2D.

Cabe señalar que el resultado obtenido para el primer método, empleando la cabeza media de BFM, se ha tomado la reconstrucción tras el proceso de refinado. Si tomásemos el MSE de la reconstrucción en la primera iteración, sería muy similar al segundo método, 1,13, según la Tabla 2.1.

MÉTODO		MSE (mm)		
DETECCIÓN PTOS.	CÁLCULO DE POSES DE LA CÁMARA	REC. - GT3D	MIN	MAX
IntraFace	POSIT + cabeza media	2,30	0,32	7,96
IntraFace	Matriz Fundamental	3,70	0,77	11,49
SURF + *tracking*	Matriz Fundamental	4,29	0,83	12,24

Tabla 5.2. MSE promedio para todos los usuarios de la base de datos sintética. Puntos de *IntraFace* promediados.

Para los puntos de *IntraFace* promediados, vemos como el MSE es mayor en el último método implementado, un 86% superior al primer método desarrollado mediante la cabeza media de BFM y POSIT para calcular las poses relativas de la cámara. En las tablas se puede comprobar que, a medida que hacemos más genérico el algoritmo, es decir, más independencia de funciones externas o cantidad de información de entrada al algoritmo de reconstrucción, empeoramos un poco la reconstrucción obtenida, pero el algoritmo se vuelve más realizable para una aplicación práctica, donde se puede disponer de cualquier algoritmo de detección de puntos.

Error en HPE.

En las siguientes tablas, se muestra el error en HPE en traslación y rotación. Los valores más importantes se corresponden al **GT zeroed - HPE zeroed** en ambos casos, ya que se debe tomar tanto en la reconstrucción como en el *ground truth*, la misma referencia para adoptarlas a la posición cero en el primer fotograma. En la práctica, resulta muy complicado que la posición inicial de la cabeza se sitúe en cero, razón para realizar dicha modificación.

MÉTODO		Error en HPE (º) en rotación		
DETECCIÓN PTOS.	CÁLCULO DE POSES DE LA CÁMARA	Absoluto	GT - HPE zeroed	GT zeroed - HPE zeroed
IntraFace	POSIT + cabeza media	3,09	3,10	0,58
IntraFace	Matriz Fundamental	2,44	2,53	0,94
SURF + *tracking*	Matriz Fundamental	2,35	2,07	1,47

Tabla 5.3. Error HPE promedio en rotación para todos los usuarios de la base de datos sintética. Puntos de *IntraFace* promediados.

MÉTODO		Error en HPE (mm) en traslación	
DETECCIÓN PTOS.	CÁLCULO DE POSES DE LA CÁMARA	Absoluto	GT zeroed - HPE zeroed
IntraFace	POSIT + cabeza media	84,33	9,94
IntraFace	Matriz Fundamental	285,18	6,45
SURF + *tracking*	Matriz Fundamental	272,63	3,59

Tabla 5.4. Error HPE promedio en traslación para todos los usuarios de la base de datos sintética. Puntos de *IntraFace* promediados.

En cuanto a los resultados del error en HPE, obtenemos resultados muy similares en todos los casos. Valores bajos de error en los dos primeros métodos, inferior a 1º en rotación. En traslación se produce el fenómeno inverso: el último método obtiene el mejor resultado. El primer método, el valor de error más elevado.

Todos estos resultados concluyen que métodos más específicos obtienen mejores resultados en la reconstrucción. El primer método está supeditado a trabajar únicamente con puntos de *IntraFace*. El último método aporta mucha libertad en la selección de puntos, pero obtiene peores resultados. Por tanto, la aplicación determina el uso de un método u otro, derivando en una mayor o menor fidelidad en la reconstrucción final.

5.2. Resumen de los resultados. Base de datos real.

MSE.

Para la base de datos real, también es posible obtener el MSE y el error en la estimación de la pose de la cabeza, pero estos errores, como ya se ha comentado anteriormente, es aproximado, ya que se adaptan los 54 puntos del GT3D a los 43 puntos de *IntraFace*, mediante la selección de puntos similares y reordenación de éstos. En esta base de datos, solo se han considerado las reconstrucciones empleando puntos de *IntraFace* promediados. Así como ocurría en la base de datos sintética, el MSE es superior para el método más general, el desarrollado en el cuarto capítulo, donde se describe todo el proceso de reconstrucción SfM.

MÉTODO		MSE (mm)		
DETECCIÓN PTOS.	CÁLCULO DE POSES DE LA CÁMARA	REC. - GT3D	MIN	MAX
IntraFace	POSIT + cabeza media	5,44	0,84	12,18
IntraFace	Matriz Fundamental	5,86	0,96	13,18
SURF + *tracking*	Matriz Fundamental	6,55	1,02	16,65

Tabla 5.5. MSE promedio para todos los usuarios de la base de datos real. Puntos de *IntraFace* promediados.

Error en HPE.

MÉTODO		Error en HPE (º) en rotación		
DETECCIÓN PTOS.	CÁLCULO DE POSES DE LA CÁMARA	Absoluto	GT - HPE zeroed	GT zeroed - HPE zeroed
IntraFace	POSIT + cabeza media	2,18	2,16	1,52
IntraFace	Matriz Fundamental	2,40	2,03	1,40
SURF + *tracking*	Matriz Fundamental	2,12	1,98	1,48

Tabla 5.6. Error HPE promedio en rotación para todos los usuarios de la base de datos real. Puntos de *IntraFace* promediados.

MÉTODO		Error en HPE (mm) en traslación	
DETECCIÓN PTOS.	CÁLCULO DE POSES DE LA CÁMARA	Absoluto	GT zeroed - HPE zeroed
IntraFace	POSIT + cabeza media	155,81	9,48
IntraFace	Matriz Fundamental	286,79	4,15
SURF + *tracking*	Matriz Fundamental	275,47	3,72

Tabla 5.7. Error HPE promedio en traslación para todos los usuarios de la base de datos real. Puntos de *IntraFace* promediados.

Para el error en HPE, se observa que el error en traslación se mantiene constante para todos los casos, pero en traslación, el error **GT zeroed- HPE zeroed** es mejor en el caso del último método, así como ocurría en la base de datos sintética.

Los resultados en la base de datos real ratifican los observados en la base de datos sintética.

5.3. Consideraciones finales.

El desarrollo de los respectivos métodos de cada capítulo demuestra la importancia de comprender la función del proceso que se realiza en cada paso para poder analizar su impacto final en la reconstrucción, puesto que existen múltiples factores que intervienen en él. Se observó que la primera aproximación del tercer capítulo era capaz de hacer mejores reconstrucciones que la segunda aproximación, pero con cierta probabilidad. El mismo fenómeno ocurría con los métodos del cuarto capítulo. Es importante procurar que prime la obtención de reconstrucciones deterministas entre ejecuciones del algoritmo, razón por la cual se han desarrollado las segundas aproximaciones de los métodos con el fin de mitigar el efecto que pueden tener funciones estadísticas como RANSAC sobre el algoritmo. Asegurar el determinismo se ha convertido en el reto del proyecto.

Los resultados, en general, son satisfactorios. Se ha comprendido el proceso completo de reconstrucción mediante SfM y se han desarrollado nuevas alternativas a la reconstrucción, como es el caso de emplear *tracking* KLT para obtener las poses de la cámara, todo ello mediante las nuevas funciones de MATLAB. Esto amplía el horizonte para seguir desarrollándose en un futuro nuevos métodos mediante la introducción de nuevas herramientas en la detección de puntos y búsqueda de correspondencias, por ejemplo, versiones póstumas de *IntraFace*, o bien, la opción de trabajar con otras bases de datos. SfM permite establecer una alternativa a la reconstrucción muy provechosa por medio de hardware y software de bajo coste. Su principal limitación es la necesidad de aportar la calibración de la cámara que ha realizado la obtención de las imágenes empleadas y la indeterminación del factor de escala en la reconstrucción, si no se aporta la matriz extrínseca.

Por otro lado, no se ha conseguido implementar un algoritmo viable en la reconstrucción mediante detección de puntos SURF y BRISK, complementado con *matching* en la descripción para la búsqueda de correspondencias. Tampoco se ha llegado a realizar la optimización de análisis de componentes principales (PCA) de modelos a partir de la reconstrucción densa, con el fin de hacer un ajuste de modelos deformables a una nube de puntos dispersa. En este último caso, serviría para contrastar los resultados del error en la base de datos sintética respecto al MSE y el error en HPE, así como la posibilidad de obtener una reconstrucción mejorada para usuarios reales.

El último método desarrollado tiene además otra gran ventaja respecto a los demás, y es la posibilidad de realizar reconstrucciones para cualquier otro tipo de objetos. Los otros métodos solo funcionan para reconstruir modelos de cabeza, por su dependencia con el algoritmo de *IntraFace*, únicamente aplicable a caras.

6. Bibliografía.

[1] R. SEGURA. Reconstrucción 3D de modelos personalizados de cabeza para la estimación de la pose. UPNA, 2015.

[2] O. FAUGERAS. Three Dimensional Computer Vision. MIT Press, 1993.

[3] M.J. WESTOBY, J. BRASINGTON, N.F. GLASSER, M.J. HAMBREY, J.M. REYNOLDS. 'Structure from Motion' photogrammetry: A low-cost, effective tool for geoscience applications. Aberystwyth University, UK, 2012.

[4] BAY, H., TUYTELAARS, T., & VAN GOOL, L. SURF: Speeded up robust features. In Computer Vision–ECCV 2006 (pp. 404-417). Springer Berlin Heidelberg, 2006.

[5] LOWE, D. G. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 2004.

[6] LEUTENEGGER, S., CHLI, M., & SIEGWART, R. Y. BRISK: Binary robust invariant scalable keypoints. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2548-2555). IEEE.

[7] P. PAYSAN, REINHARD KNOTHE, B. AMBERG, S. ROMDHANI, T. VETTER. A 3D Face Model for Pose and Illumination Invariant Face Recognition. Genova, Italia, 2009.

[8] C. ORDOYO. Plataforma de simulación para algoritmos de estimación de la posición 3D: estudio de variabilidad. UPNA, 2014.

[9] J. J. BENGOECHEA. Comparativa de algoritmos de visión monocular para la estimación de la posición de la cabeza. UPNA, 2014.

[10] X. XIONG, F. DE LA TORRE. Supervised Descent Method and its Application to Face Alignment. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[11] HARTLEY, R. AND A. ZISSERMAN. Multiple View Geometry in Computer Vision. Cambridge University Press, 2003.

[12] M. GAMALLO. Memoria final de Beca de Colaboración. UPNA, 2014.

[13] RICHARD I. HARTLEY. In Defence of the 8-point Algorithm. GE-Corporate Research and Development, Schenectady, NY.

[14] M.A. FISCHLER, R.C. BOLLES. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395, 1981.

[15] S. CHOI, T. KIM, W. YU. Performance Evaluation of RANSAC Family. Robot Research Department, Daejeon, Korea, 2009.

[16] LUCAS, BRUCE D. AND TAKEO KANADE. An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence, April, 1981, pp. 674–679.

[17] A. GÓNZALEZ. Seguimiento y Producción Automática mediante Cámaras PTZ en Entornos de Red. Univ. Autónoma de Madrid, 2013, pp. 24-25.

[18] KALAL, ZDENEK, KRYSTIAN MIKOLAJCZYK, AND JIRI MATAS. Forward-Backward Error: Automatic Detection of *Tracking* Failures. Proceedings of the 20th International Conference on Pattern Recognition, 2010, pp. 2756–2759, 2010.

[19] P.J. BESL, H.D. MCKAY. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992.