Universidad Zaragoza
1542

Trabajo Fin de Máster

Structure from Motion deformable en secuencias de endoscopia médica

Deformable Structure from Motion in medical endoscopy sequences

Autor

Iñigo Cirauqui Viloria

Director

José María Martínez Montiel

Escuela de Ingeniería y Arquitectura

2022

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./Dª. Íñigo Cirauqui Viloria ,

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de Máster Universitario en Ingeniería Industrial (Título del Trabajo)

Structure from Motion deformable en secuencias de endoscopia médica

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 25 de Noviembre de 2022

Fdo: Íñigo Cirauqui

Agradecimientos

Este proyecto ha recibido financiación del programa de Investigación e Innovación Horizon 2020 de la Unión Europea, bajo el acuerdo de subvención número 863146.

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 863146

Resumen

Structure from Motion deformable en secuencias de endoscopia médica

Las técnicas de Structure from Motion (SfM) son empleadas para obtener representaciones 3D de escenas observadas en una secuencia de vídeo o conjunto de imágenes. El algoritmo que las gobierna logra esto al ser capaz de identificar puntos comunes entre las diferentes imágenes, triangular estos en el espacio y calcular la posición de las cámaras respecto de ellos. Para conseguirlo asume que la escena que está observando es rígida y que cada punto a triangular mantiene constante su posición relativa respecto de los demás.

Al intentar emplear estos algoritmos para procesar escenas que contienen cuerpos deformables como aquellos presentes en secuencias médicas, la eficacia del algoritmo desciende rápidamente conforme aumenta la deformación del tejido. Modificar los algoritmos para permitir su funcionamiento en entornos deformables requiere identificar y caracterizar esa deformación.

En el proyecto se han desarrollado métodos para potenciar el rendimiento del SfM en escenas de colonoscopia. La secuencia completa se divide en trozos de corta duración dónde la deformación es lo bastante reducida como para que funcione el SfM, permitiendo construir modelos mecánicos de la escena deformada. Al comparar estos modelos con el que resulta de procesar la secuencia completa se logra medir el efecto de la deformación.

Summary

Deformable Structure from Motion in medical endoscopy sequences

Structure from Motion is an algorithm to acquire 3D representations of scenes captured via a video sequence or set of images; this is achieved by identifying corresponding points between the images for later triangulating them in 3D space. To do so, it assumes that the observed scene is rigid and that the relative position of each point to the rest remains constant.

When these algorithms are applied to scenes containing deformable bodies, such as medical images, the algorithm's performance rapidly drops as the deformation rises. Modifying these algorithms to allow their process in deformable environments requires identifying and measuring this deformation.

Through this project, we've developed new methods for enhancing SfM capabilities in colonoscopy procedures. The sequence is split into short videos in which the accumulated deformation is small enough to pass through the SfM, allowing mechanical models of the deformable scene to be built. By comparing these models with the one resulting from processing the complete sequence, the magnitude of the deformation can be measured.

Índice

Capítulo 1

Introducción

Se denomina **Structure from Motion** (SfM), o estructura a partir del movimiento, a un conjunto de algoritmos que permiten aproximar una estructura tridimensional a partir de imágenes de la misma tomadas con diferente posición relativa al objeto, generando una nube de puntos que lo representa y calculando la posición y orientación de las cámaras respecto del mismo. Esta técnica puede extrapolarse de objetos a entornos completos, pudiendo por ejemplo obtener una reconstrucción tridimensional de un edificio o una ciudad.

El SfM tiene sus orígenes en el campo de la visión por computador y el desarrollo de algoritmos de correlación automática de imágenes. El término fue empleado por primera vez por Ullman [1], quien en su aplicación y considerando cámaras ortogonales, obtiene la forma de un objeto rígido a partir de varias imágenes. Generalizando esto, la misma reconstrucción se llevaba a cabo con diferentes vistas de un objeto en movimiento. La técnica comienza a emplearse para fotogrametría en Bolles et al. [2] donde se desarrolla un método iterativo para obtener la posición de cinco puntos a partir de dos imágenes. Más recientemente, cabe destacar los trabajos de Hartley y Zisserman [3]; y Schönberger [4], que combinan diversos métodos para mejorar el proceso en aspectos como la selección de imágenes y la triangulación de puntos. En la actualidad algunas de las principales aplicaciones del algoritmo son la reconstrucción de escenas y la fotogrametría, siendo muy preciso en esta última.

La **endoscopia** es una técnica diagnóstica basada en la introducción de una única cámara, el endoscopio, a través de un orificio natural o provocado para visualizar una cavidad corporal y recopilar información sobre ella. La colonoscopia es una particularización de lo anterior cuyo procedimiento requiere recorrer el tracto intestinal detectando anomalías en el tejido para en ocasiones interactuar con ellas. El limitado movimiento del endoscopio induce en el equipo médico una dificultad para orientarse, lo cual es necesario si se ha de volver a estudiar la anomalía en una intervención posterior, o si la posición se ha de alcanzar de nuevo con herramientas adicionales.

El SfM aplicado a este entorno puede permitir la obtención de un mapa tridimensional de la cavidad y la posición del endoscopio en la misma, ayudando a los equipos médicos en el procedimiento.

1.1. Motivación y objetivos

El método de reconstrucción empleado en el SfM se basa en reconocer puntos comunes entre las imágenes que observan el mismo entorno para después triangularlos en el espacio; el algoritmo considera que la escena es rígida, es decir, que la posición relativa de cada punto triangulado con respecto a los demás no varía y es observable desde diferentes posiciones.

Al intentar aplicar el mismo método a una secuencia médica *in vivo*, surge la dificultad de que el tejido está sometido a constante deformación, causando que la posición relativa entre los mismos puntos cambie con el tiempo, dificultando o impidiendo la reconstrucción.

El alcance de nuestra contribución tiene dos objetivos:

1. Extender el funcionamiento de un algoritmo de SfM para procesar escenas médicas, permitiendo obtener reconstrucciones de las mismas que incluyan posiciones deformadas.

2. Identificar y caracterizar la deformación existente en la escena mediante modelos mecánicos.

1.2. Estructura

El capítulo 2 introduce el algoritmo de SfM y las particularidades de las lentes ojo de pez (*fish-eye*) instaladas en los endoscopios, así como un nuevo método de búsqueda guiada de puntos correspondientes diseñado para estas lentes.

El capítulo 3 presenta un método de post-procesado de los datos del SfM diseñado para medir la deformación de las escenas que se basa en construir modelos de elementos finitos a partir de las imágenes.

El capítulo 4 valida el método con una secuencia de endoscopia *in vivo*.

El capítulo 5 aglutina las conclusiones y propone líneas de trabajo futuro.

Capítulo 2

Geometría epipolar en las cámaras fish-eye

Los algoritmos de visión por computador empleados en el SfM basan su funcionamiento en reconocer puntos comunes entre un conjunto de imágenes del mismo entorno, pudiendo estas imágenes ser parte de una secuencia de vídeo. Al observar el mismo punto desde varias posiciones es posible triangular su posición en el espacio por intersección de rayos proyectantes, esto es, el rayo que une el centro óptico de la cámara con el punto en la imagen como se muestra en la Figura 2.1a. Realizando esto con todos los puntos correspondientes se obtiene una representación de la escena en forma de nube de puntos, así como la posición y orientación de las cámaras respecto de ella.

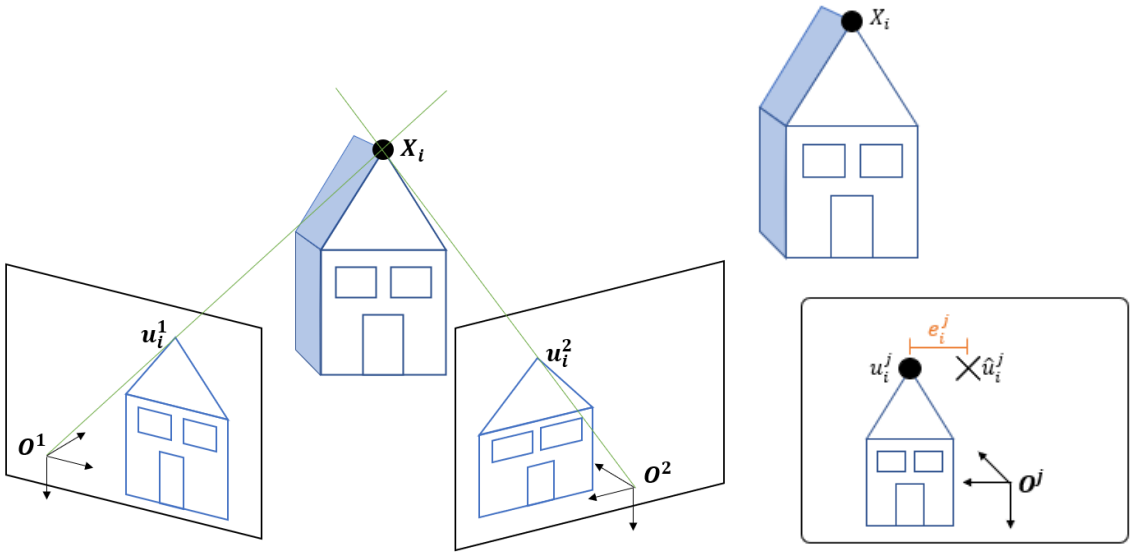
Para realizar este proceso el algoritmo asume que el entorno procesado es **rígido**, es decir, que la posición relativa entre los puntos que se están triangulando no varía. Las secuencias médicas contienen tejido vivo sujeto a **deformación** lo que dificulta la actuación del SfM, que solo logra obtener una aproximación de la escena.

Planteamos la hipótesis de que generando mapas en intervalos de tiempo lo bastante cortos, por ejemplo con las imágenes contenidas en 0.5 segundos de vídeo, la deformación del tejido será lo bastante pequeña como para que el algoritmo funcione correctamente, permitiendo mapear el tejido deformado. En el presente trabajo desarrollamos herramientas para capturar y analizar esta deformación buscando facilitar el funcionamiento de los algoritmos de Visión por Computador en entornos deformables.

2.1. Structure from Motion

Hemos iniciado el estudio empleando el software de SfM COLMAP [4, 5], cuyo proceso de construcción de mapas es el siguiente:

1. **Detección de puntos característicos:** se dispone de un conjunto de imágenes mostrando el mismo entorno en las que se identifican puntos característicos empleando un detector de SIFT (Scale-Invariant-Feature-Transform) [6]; para cada punto se utiliza su entorno cercano (píxeles próximos) con el fin de calcular una firma única que lo describe, permitiendo reconocerlo en otras imágenes.



(a) (b)

Figura 2.1: (a) Si dos cámaras observan el mismo punto (u_i^1 y u_i^2), cada una de ellas con posición y orientación conocidas (θ_{ext}^1 y θ_{ext}^1), con los parámetros que definen las características de las lentes también conocidos (θ_{int}^1 y θ_{int}^1), pueden obtenerse las coordenadas del punto en el espacio (X_i) por intersección de rayos proyectantes, siendo estos los rayos que pasan por el centro óptico de la cámara (O^1 y O^2) y el punto en la imagen, representados en verde. (b) El rectángulo representa una imagen en la que se observa un objeto (casa), del que se conocen las coordenadas de un punto 3D X_i. Al reproyectar el punto 3D sobre la imagen j se obtienen sus coordenadas en $\hat{u}_i^j$, sin embargo, el punto en la imagen se encuentra en las coordenadas u_i^j; la distancia entre la posición del punto proyectado y la posición de la imagen es el error de reproyección e_i^j del punto i en la cámara j.

2. **Búsqueda de correspondencias:** con las imágenes analizadas, cada punto de cada imagen se compara contra los puntos de las demás, considerando que dos de ellos son correspondientes si sus firmas difieren menos que un umbral.

3. **Inicio de la reconstrucción:** en esta etapa se inicializa la nube de puntos a partir de dos imágenes. Para ello se emplea RANSAC (RANdom SAmple Consensus) [7] para seleccionar un subconjunto de puntos correspondientes con los que construir una matriz esencial **E**, que define la transformación geométrica entre ambas cámaras y, a partir de la cual, se estima su posición 3D.

4. **Reconstrucción incremental:** partiendo de la inicialización anterior se añaden al mapa el resto de imágenes, posicionándolas respecto de las primeras y triangulando nuevos puntos. Durante el proceso se llevan a cabo sucesivas etapas de ajuste de haces o Bundle Adjustment (BA) [8] con función de influencia robusta, una optimización no lineal que varía conjuntamente la pose (posición y orientación) estimada para las cámaras y las coordenadas 3D de los puntos, buscando minimizar el error de reproyección, que para un punto se define como la distancia

entre la posición proyectada del punto 3D sobre la imagen, con la posición real del punto detectado en la imagen, mostrando un ejemplo en la Figura 2.1b.

Tras procesar todas las imágenes se obtiene una reconstrucción como la señalada en la Figura 2.2. En ella las pirámides rojas representan las posiciones de la cámara calculadas para cada imagen de la secuencia de vídeo, en torno a las que se encuentran los puntos triangulados.

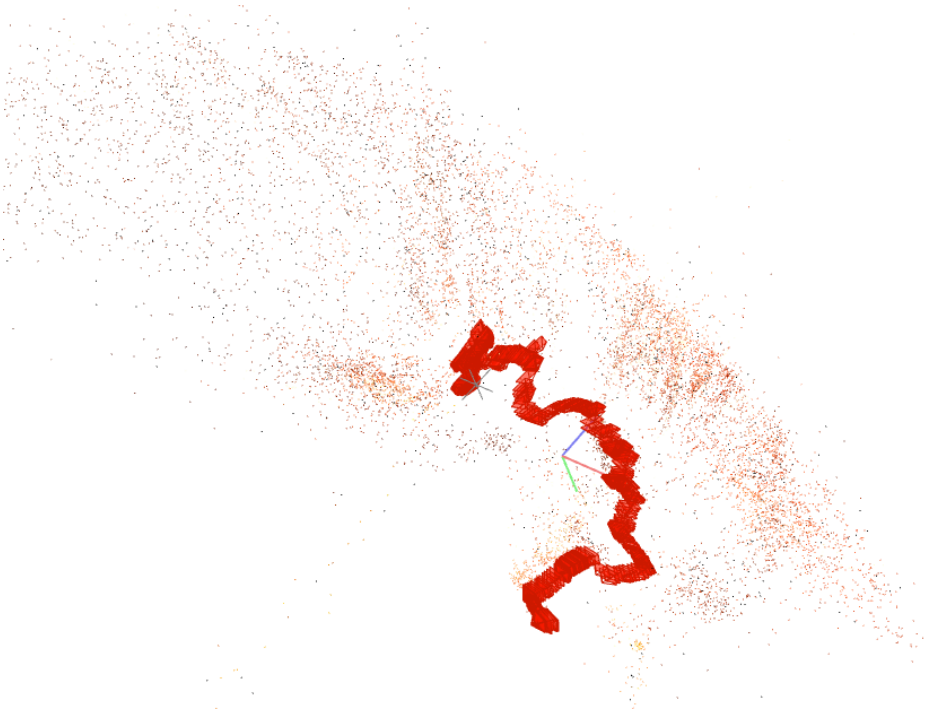


Figura 2.2: Ejemplo de reconstrucción resultado del proceso de SfM: las pirámides rojas corresponden con las posiciones de las imágenes que forman la secuencia de vídeo y que juntas forman la trayectoria seguida por la cámara; alrededor de ellas se encuentran los puntos triangulados a partir de varias observaciones en las imágenes.

En las etapas de búsqueda de correspondencias, inicialización y reconstrucción incremental, se realiza un proceso de verificación geométrica de los resultados basado en una búsqueda guiada de correspondencias 3D-2D, el cual emplea un modelo matemático de las cámaras para proyectar los puntos 3D sobre las imágenes y poder medir el error de reproyección:

1. El modelo de cámara se define con sus parámetros intrínsecos θ_{int} que definen las características de la lente (posición de su centro óptico, distancia focal, distorsión) y los parámetros extrínsecos θ_{ext} que definen su pose (posición y orientación).

2. Conocida la pose de varias cámaras y un conjunto de puntos triangulados, estos se reproyectan sobre las imágenes empleando la Ecuacion 2.1, obteniendo su posición en el plano de imagen. En la ecuación, P^j representa la matriz de proyección de la cámara j, que se construye como combinación de K^j (matriz de calibración

obtenida de los parámetros intrínsecos mostrada en la Ecuación 2.2) y $[R^j|t^j]$ matriz de posición y orientación obtenida de los parámetros extrínsecos.

$$\hat{u}_i^j = P^j(X_i, \theta_{int}^j, \theta_{ext}^j) = P^j X_i = K^j[R^j|t^j]X_i \tag{2.1}$$

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

3. Se mide la distancia entre la *proyección* del 3D y el punto de la imagen que se ha empleado para construirlo.

4. Esta distancia corresponde con el error de reproyección y ha de ser menor que un umbral para considerar válida la asociación entre el punto de imagen y el 3D.

5. Si el error obtenido con todas las observaciones (reproyección del 3D en todas las imágenes con puntos que lo triangulan) fuese demasiado grande se consideraría que en realidad el punto ha sido mal triangulado o que los puntos empleados para posicionarlo en el espacio no son correspondientes.

El cálculo de este error de reproyección se expresa en la Ecuación 2.3. La posición 3D de los puntos (X_i) y la pose de las cámaras (θ_{ext}^j) se modifican en mediante un proceso de BA con función de influencia robusta, que minimiza la función de coste expresada en la Ecuación 2.4, en cuyo mínimo se encuentra la posición óptima de cámaras y puntos, que debería ser la que mejor representa la realidad.

$$e_r^j = \sum_{i,j} \left(u_i^j - P^j\left(X_i, \theta_{ext}^j, \theta_{int}^j\right)\right)^2 \tag{2.3}$$

$$\underset{\mathbf{X}_i, \theta_{ext}^j}{\arg\min} \sum_{i,j} \rho\left(\left(u_i^j - P^j\left(X_i, \theta_{int}^j, \theta_{ext}^j\right)\right)^2\right) \tag{2.4}$$

Además de eliminar emparejamientos erróneos entre puntos de imagen, el proceso anterior también impone una condición de rigidez a la escena: si un punto ha sido triangulado a partir de imágenes en el instante t_0 de la secuencia y posteriormente se observa con imágenes en un instante posterior t_1, habiéndose deformado la escena entre ambos instantes, el cambio ocurrido impondrá un error de reproyección muy grande para ese punto 3D, causando que sea descartado.

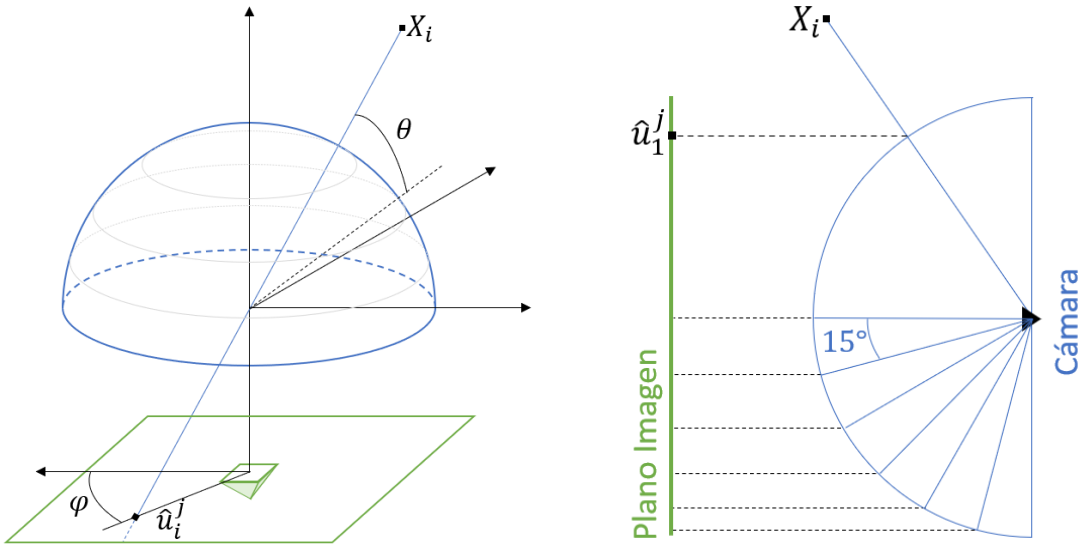
Al eliminar los puntos sometidos a deformación considerable, la reconstrucción obtenida tras procesar la secuencia de vídeo corresponde con una nube de puntos que representa la envolvente rígida de la escena, esto es, aquellos puntos con un movimiento relativo lo bastante mínimo como para ser considerados rígidos por los umbrales del SfM, no habiendo recabado información de la escena en posiciones deformadas. En el proyecto se añaden las estructuras necesarias para solventar esto.

2.2. Búsqueda de correspondencias en las cámaras fish-eye

Las cámaras de los endoscopios del estudio están equipadas con lentes de ojo de pez (*fish-eye*), cuya geometría se define con un modelo de cámara Kannala-Brandt [9]. Para calcular la posición de cada punto 3D en la imagen se emplean las Ecuaciones 2.5 y 2.6, en las que θ es la inclinación del rayo incidente y φ el azimut del mismo, representados en la Figura 2.3a. De acuerdo con estas ecuaciones, al proyectar la imagen sobre un plano, la distorsión a aplicar sobre los píxeles aumenta conforme nos alejamos del centro de la imagen como se muestra en la Figura 2.3b.

$$\begin{pmatrix} x \\ y \end{pmatrix} = r(\theta) \begin{pmatrix} cos\varphi \\ sin\varphi \end{pmatrix} \tag{2.5}$$

$$r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^7 \tag{2.6}$$



(a) (b)

Figura 2.3: Geometría de lente para cámaras fish-eye. (a) Tomando un punto X_i sobre una cámara fish-eye (azul), se muestra su posición en el plano de imagen (verde) en función del ángulo *theta* que mide la inclinación respecto al plano de imagen y el ángulo φ que mide el azimut. (b) Muestra una lente ojo de pez en dos dimensiones, las líneas mostradas, que se encuentran espaciadas 15 grados unas de otras, muestran cómo las proyecciones en el plano de imagen se juntan conforme nos acercamos al borde de la imagen. De este modo, los puntos vistos por el centro de la cámara sufren menos distorsión que los vistos en el contorno, como es el caso de X_i.

Para minimizar el impacto de la distorsión se modifica el proceso de búsqueda guiada de correspondencias que, para un modelo simplificado de dos imágenes, pasa a seguir el siguiente procedimiento apoyado en la Figura 2.4:

1. Aplicando la geometría epipolar cada punto x de la imagen 1 define una línea epipolar l' en la imagen 2; así mismo cada punto x' de la imagen 2 define una línea epipolar l en la imagen 1. Las líneas se calculan mediante las Ecuaciones 2.7, donde F es la Matriz Fundamental fruto de combinar la matriz de calibración K (obtenida a partir de los parámetros intrínsecos de la cámara) mostrada en la Ecuación 2.2 con la Matriz Esencial E, que define la transformación entre las imágenes, como se muestra en la Ecuación 2.8.

$$\begin{aligned} l' &= Fx \\ l &= F^T x' \end{aligned} \tag{2.7}$$

$$F = K^{-T} E K^{-1} \tag{2.8}$$

2. Los puntos de la imagen 2 candidatos a ser correspondientes con el de la 1 se encontrarán en las proximidades de esa línea. En el procedimiento original se define una región de búsqueda en torno a esta como muestra la Figura 2.5a y se busca un punto de imagen correspondiente dentro de esta banda.

3. Empleando la ecuación de la recta epipolar en la imagen 2 representada en la Ecuación 2.9 y conocida la posición del centro óptico de la cámara O_2, obtenido a partir de los parámetros extrínsecos de esta y definido en la Ecuación 2.10; se calcula el plano epipolar π que contiene a la recta y al centro óptico y se define con la Ecuación 2.11.

$$l_0 x + l_1 y + l_2 = 0 \tag{2.9}$$

$$O_2 = (c_x, c_y, f) \tag{2.10}$$

$$\pi_0 x + \pi_1 y + \pi_2 z + \pi_3 = 0 \tag{2.11}$$

4. A continuación, se define el rayo de vista que une el centro óptico de la segunda cámara O_2 con el punto candidato en la imagen x', rayo representado en azul en la Figura 2.5b.

5. Por último, se obtiene el ángulo φ que forma dicho rayo con el plano epipolar, para lo que de acuerdo al procedimiento descrito en [10] y asumiendo ángulos pequeños, se emplea la Ecuación 2.12 en la que $u_{O_2x'}$ es el vector director del rayo que une el centro óptico con el punto en la imagen y n_π el vector normal del plano epipolar.

$$\varphi_{x'} = \frac{|x' \cdot l'|}{\|u_{O_2x'}\| \|n_\pi\|} \tag{2.12}$$

6. El valor de φ representa el error angular para el punto candidato a ser correspondiente. Solo aquellos puntos de la imagen 2 que resulten en un error menor que un umbral establecido en un grado son candidatos a ser correspondientes con el punto original.

7. Se comparan el descriptor del punto de la imagen 1 con los descriptores de los puntos candidatos en la imagen 2, seleccionando aquel con menor distancia de *Sampson*, que mide la diferencia entre los descriptores y se define en la Ecuación 2.13.

$$d = \sqrt{\frac{(x'^T F x)^2}{l_0^2 + l_1^2 + l_0'^2 + l_1'^2}} \tag{2.13}$$

Esta modificación aumenta el número de correspondencias encontradas, así como el número de observaciones para cada punto triangulado.

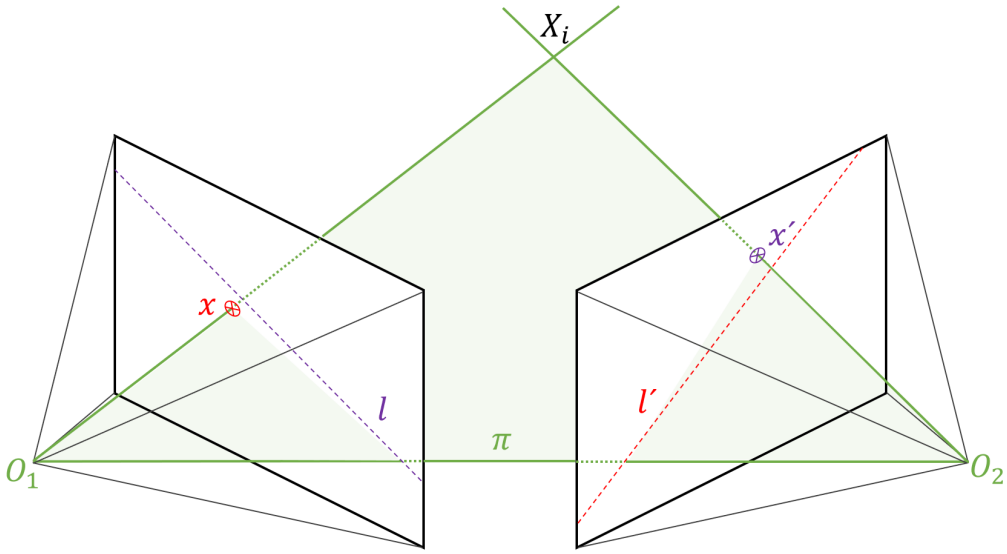
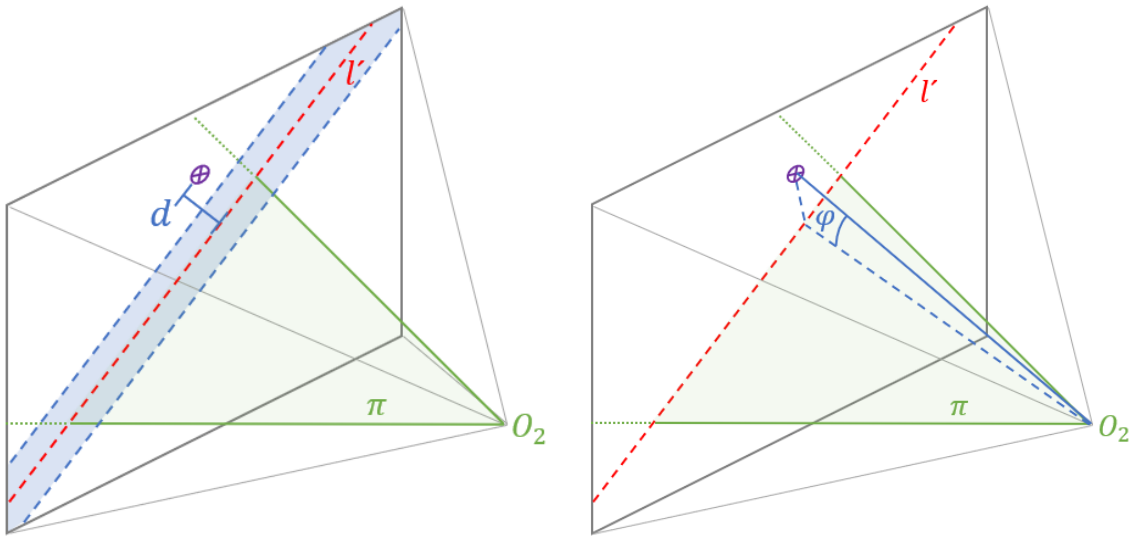


Figura 2.4: Geometría epipolar. Se muestra cómo un punto x de la imagen 1 se corresponde con una línea l' en la imagen 2 y viceversa para el par [x', l]; el plano epipolar π contiene los centros ópticos de las cámaras y el punto 3D, situado en la intersección de los rayos proyectantes.

2.3. Registro de puntos eliminados en SfM

Cuando los puntos con elevado error de reproyección son eliminados, consideramos que esto es debido a que no tienen un comportamiento rígido y varían su posición en el tiempo causando que la escena vista no se corresponda con el mapa generado a partir de imágenes anteriores.

Para cada punto eliminado, hemos recabado información sobre su posición 3D y qué puntos de imagen se han empleado para triangularlo, posteriormente hemos dibujado



(a) (b)

Figura 2.5: (a) En el método original de emparejamiento se define en el plano de la imagen 2, una región de búsqueda en torno a la línea epipolar l', un punto es candidato a ser correspondiente si la distancia d es menor que el umbral establecido por la región de búsqueda. (b) En el nuevo método, tras obtener la recta epipolar, se calcula una línea entre el centro óptico de la cámara y el punto de imagen candidato a ser correspondiente con el de la imagen 1, estableciendo un umbral en el ángulo φ que esta línea forma con el plano epipolar.

estos resultados sobre las imágenes de la secuencia con el objetivo de identificar regiones con elevada deformación. Así, por ejemplo, en la Figura 2.6 pueden verse los puntos verdes que corresponden a la proyección de puntos 3D activos y los rojos a puntos 3D que han sido eliminados por no cumplir la condición de rigidez. Finalmente se dibujan en gris el resto de puntos de imagen no empleados en ninguna triangulación.

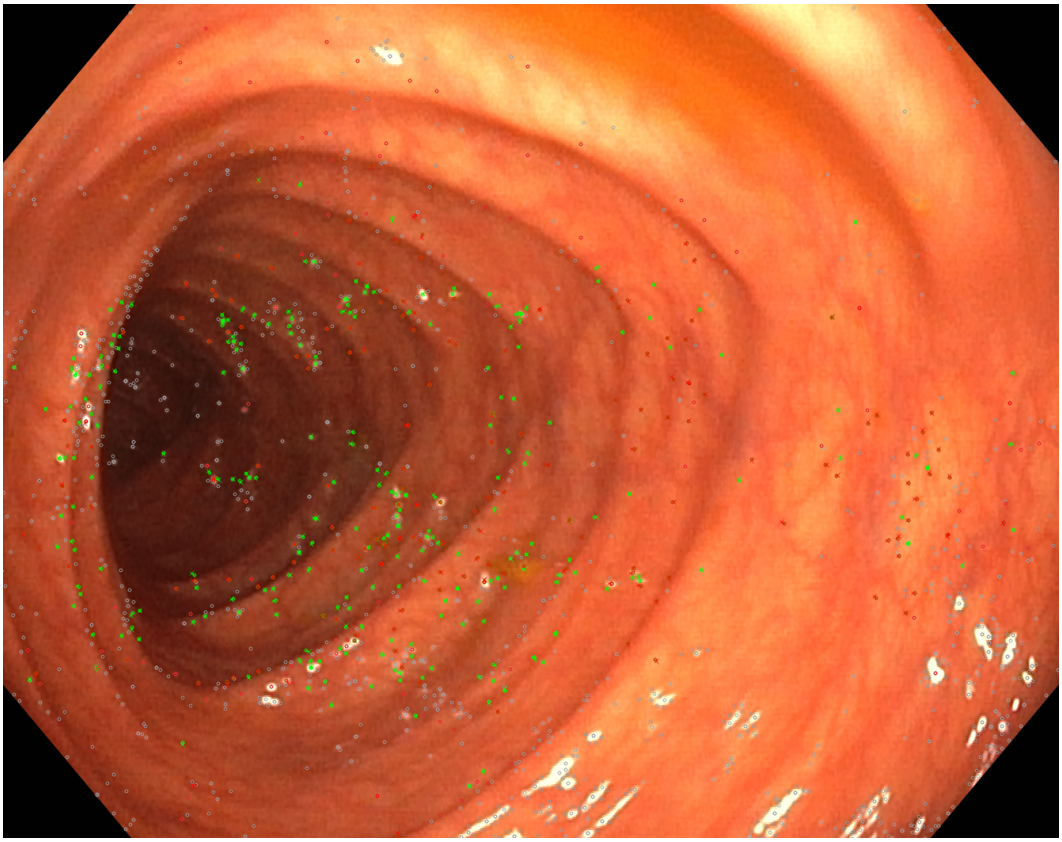


Figura 2.6: Los puntos verdes representan aquellos puntos de imagen asociados con un punto 3D presente en el mapa. Los rojos representan aquellos que se usaron para triangular un punto que se han eliminado por no cumplir la condición de rigidez. Los puntos grises representan el resto de puntos de imagen no empleados en el cálculo de ningún punto 3D. La elipse negra indica una zona con deformación elevada en la que se aprecia mayor concentración de puntos rojos.

Capítulo 3

Estimación de deformaciones

Al procesar una secuencia de cierta longitud, por ejemplo 10 segundos, se obtiene una reconstrucción en la forma de una nube de puntos de comportamiento rígido, es decir, estos puntos han sido observados en la misma posición desde un número suficiente de imágenes para considerarlos estáticos, lo que implica que aunque estén sometidos a deformación, esta es lo bastante pequeña para que sea soportada por los umbrales del SfM.

Otros de los puntos que se han triangulado en una superficie que se deforma en mayor medida son eliminados del mapa al no poder volver a ser encontrados en imágenes sucesivas. El siguiente ejemplo se representa en la Figura 3.1:

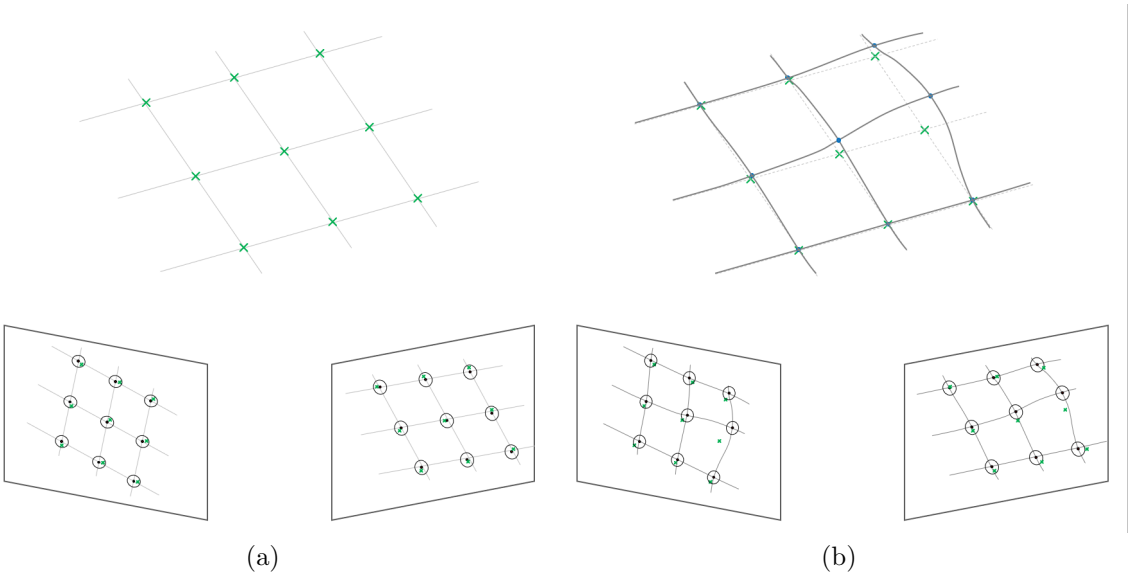
1. Tenemos un par de imágenes extraídas de la secuencia en el instante t_0 que observan una superficie, representada por las líneas grises de la mitad superior, sobre la que se han triangulado varios puntos 3D. La misma escena se ve en las imágenes de la mitad inferior y la proyección de esos puntos 3D se encuentra próxima a los puntos correspondientes en estas (dentro de las regiones de búsqueda/círculos).

2. Unos segundos después, en el instante t_1, la cámara vuelve a pasar por la misma superficie y captura un segundo par de imágenes. Sin embargo, la escena (líneas grises) se ha deformado y los puntos triangulados anteriormente (X verdes) ya no se encuentran sobre la superficie, pues el SfM asume que son rígidos y no varía su posición.

3. Cuando se proyectan esos puntos 3D sobre las nuevas imágenes que observan la escena deformada, la proyección de aquellos puntos asociados a zonas de elevada deformación se encuentra fuera de la región de búsqueda de los puntos de imagen, causando que los puntos 3D sean eliminados.

La diferencia de tiempo entre t_0 y t_1 permite que el tejido se deforme, causando que se descarten puntos. Sin embargo, construyendo un mapa solo con los frames temporalmente próximos a t_0 o t_1, la deformación es lo bastante pequeña como para que muchos de los puntos no se eliminen; permitiendo obtener mapas pequeños que representan la escena deformada.

Nuestro método para detectar la deformación consiste en construir submapas locales compuestos por unos pocos frames, de modo que el tejido no sufra una deformación excesiva que elimine parte de los puntos triangulados. Comparamos estos submapas, que representan la escena deformada, con el mapa global, que representa la estructura rígida subyacente; midiendo la diferencia para caracterizar la deformación.



(a) (b)

Figura 3.1: (a) Las líneas grises de la mitad superior representan una superficie plana, la cual es observada desde las dos cámaras de la mitad inferior; asimismo sobre la superficie se encuentran 9 puntos 3D calculados en procesos anteriores, indicados con $\times$ verdes. En las cámaras que observan la escena se han detectado puntos de imagen que aparecen en negro $\bullet$, al proyectar los 3D sobre las cámaras, cada uno de ellos se encuentra próximo al punto de imagen del que es correspondiente. En torno a cada punto de imagen se ha definido una *región de búsqueda* $\odot$ dentro de la cual ha de encontrarse el 3D proyectado. (b) En un instante posterior de la secuencia la superficie se ha deformado y los puntos 3D calculados anteriormente ya no se encuentran sobre esta. Cuando proyectamos estos antiguos 3D sobre las nuevas cámaras, que también observan la escena deformada sobre la que se han detectado puntos de imagen correspondientes, la posición proyectada se encuentra fuera de las regiones de búsqueda. Si esto sucede en un número suficiente de imágenes para el mismo el punto 3D, este será eliminado, pues se considera que no cumple con la condición de rigidez.

3.1. Mapa global y submapas locales

Cada secuencia de vídeo se procesa en dos etapas:

1. En la primera se realiza una reconstrucción con el total de las imágenes, obteniendo un mapa global que incluye todos los puntos que sobreviven tras aplicar las condiciones de rigidez en el SfM.

2. En la segunda se construyen múltiples mapas a partir de los frames de la secuencia contenidos en intervalos de 0.5 segundos. Se establecen 5 imágenes de separación entre el comienzo de cada lote de imágenes como representado en la Figura 3.2. Cada uno de estos mapas muestra una reconstrucción de la superficie local en un instante concreto.

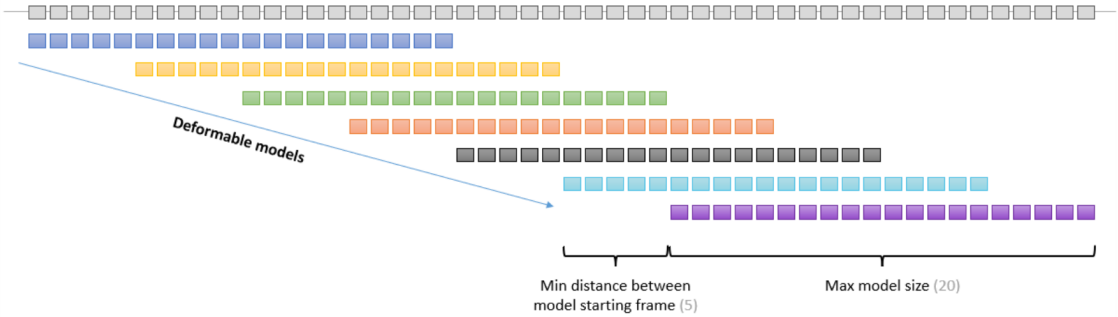


Figura 3.2: Cada pequeño cuadro representa una de las imágenes que componen el vídeo: la primera línea de cuadros grises representa todas las disponibles que se han empleado para construir el mapa global. Las líneas siguientes muestran lotes de 20 imágenes utilizadas para construir mapas locales.

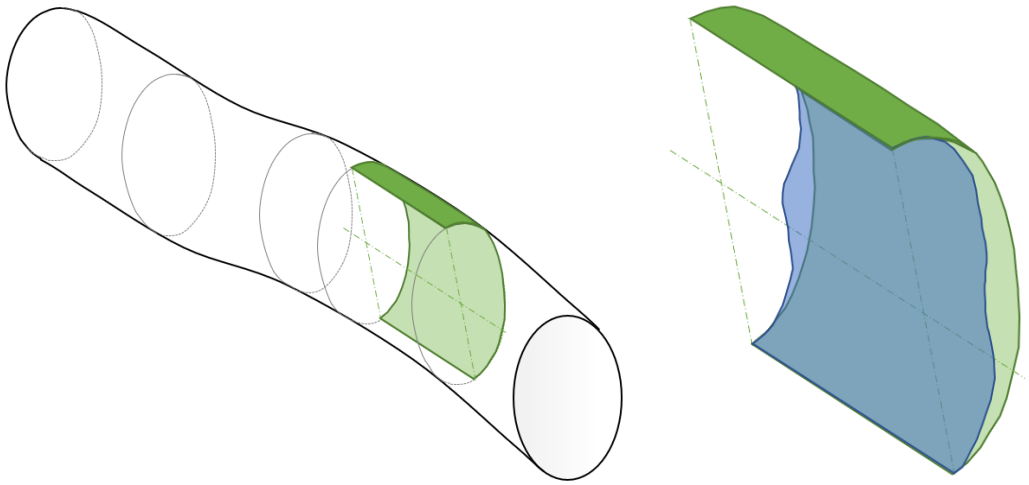
3.2. Alineación mapa global-submapa local

El mapa global muestra la estructura rígida subyacente a la escena, como estamos procesando secuencias de endoscopia, podemos aproximar esto por un cilindro como el mostrado en la Figura 3.3; cada uno de los submapas locales muestra porciones de esa escena en instantes deformados.

Para medir la deformación es necesario alinear cada uno de los submapas locales con el global. Cada mapa, está compuesto por N puntos y M imágenes, para las que se ha calculado su pose ($[R|t]$) durante el SfM. Siendo que las imágenes tienen identificadores únicos de acuerdo con su posición en la secuencia de vídeo, la alineación comienza identificando las imágenes comunes entre el mapa global y el submapa local así, por ejemplo, para cada uno de los dos submapas mostrados en la Figura 3.4, se encuentran los frames correspondientes en el mapa global.

Nuestro proceso fija en el espacio el mapa global (subíndice G) y modifica translación, rotación y escala del local (subíndice L) para alinearlo con el primero. El procedimiento selecciona una de las imágenes comunes como punto de anclaje y realiza las siguientes transformaciones:

- **Translación:** el submapa local se desplaza hacia el global hasta anular la distancia entre la imagen de anclaje en ambos modelos: $t_L = t_G$.
- **Rotación:** el submapa local se rota sobre la imagen de anclaje hasta anular la diferencia de orientación: $R_L = R_G$.



(a) (b)

Figura 3.3: (a) La estructura rígida subyacente al intestino puede aproximarse por un cilindro. En él seleccionamos una porción para la que hemos construido un submapa local. (b) El submapa local, representado en azul, muestra una reconstrucción deformada de la porción correspondiente de mapa global, representada en verde.

– **Escala:** Cada mapa se ha construido con procesos de BA independientes, con lo que sus escalas son diferentes y es necesario calcular una transformación:

  1. Se selecciona iterativamente un par de imágenes (i,j) del mapa global, midiendo la distancia entre ambas $d_G^{i,j} = t_G^i - t_G^j$. Se toma la misma medida entre las mismas imágenes en el submapa local $d_L^{i,j} = t_L^i - t_L^j$.
  2. Se calcula una escala para ese par de imágenes $s_{i,j} = d_G^{i,j}/d_L^{i,j}$.
  3. Tras obtener escalas para todos los pares de imágenes disponibles, se obtienen su media μ_s y desviación típica σ_s.
  4. Se descartan aquellas escalas superiores a tres desviaciones típicas ($s_{i,j} > 3\sigma_s$). La media de las restantes se toma como una aproximación válida de la escala entre ambos modelos.

Tras la alineación se calcula el error de reproyección acumulado entre ambos mapas, para lo que se utilizan únicamente los puntos 3D presentes en ambos. Se considera que un punto 3D es visible en los dos mapas si en cada uno de ellos ha sido triangulado a partir de las mismas imágenes.

Los puntos 3D del modelo global se proyectan sobre las imágenes del modelo local y viceversa; empleando para ello la Ecuación 2.1 del Capítulo 2. Del mismo modo para cada punto se obtiene un error de reproyección como la distancia entre su posición proyectada y la posición del punto de imagen correspondiente que se usó para triangularlo. Resolviendo la ecuación para todos los puntos e imágenes disponibles se obtiene un error de reproyección global que es indicativo de la precisión de la alineación y se

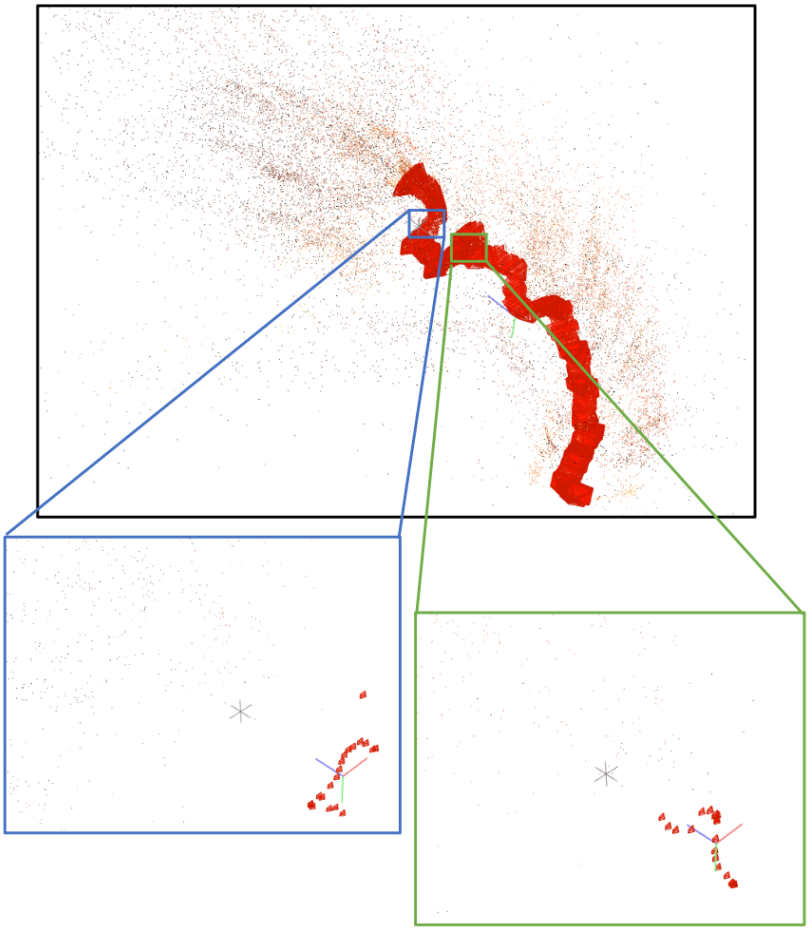


Figura 3.4: El recuadro superior muestra el mapa global y los inferiores muestran dos submapas locales, que se corresponden con las posiciones indicadas en la trayectoria global.

representa en la Ecuación 3.1, este error incluye la proyección de los puntos del mapa global sobre las imágenes del local y la proyección de los puntos del mapa local sobre las imágenes del global.

$$E_r = \sum_{i,j} \left(u_i^j - P^{j,L}\left(X_i^G, \theta_{ext}^{j-L}, \theta_{int}^{j-L}\right)\right)^2 + \sum_{i,j} \left(u_i^j - P^{j,G}\left(X_i^L, \theta_{ext}^{j-G}, \theta_{int}^{j-G}\right)\right)^2 \quad (3.1)$$

El mismo procedimiento se repite seleccionando diferentes imágenes como punto de anclaje. Aquella configuración que permita obtener el menor error de reproyección se utilizará para cálculos posteriores.

Aunque este método de alineación pudiera ser correcto para una escena rígida, la presencia de deformación disminuye la precisión debido a que el error de reproyección que está siendo calculado al proyectar los puntos del submapa local sobre el mapa global no es solo debido a una imprecisión en el momento de triangular los puntos, sino que también se ve afectado por la deformación presente en el submapa local.

Para mejorar la precisión de la alineación obtenida empleamos una modificación del proceso de Bundle Adjustment (BA):

– En el procedimiento estándar de BA, se modificarían poses de cámara y coordenadas de puntos 3D para minimizar el error de reproyección calculado en la Ecuación 3.1. En nuestra implementación, las poses y puntos del modelo rígido se mantienen fijos en el espacio, solo permitiendo el movimiento de poses y puntos correspondientes al modelo deformado como indicado en la Ecuación 3.2.

$$\underset{\mathbf{x}_i^L, \theta_{ext}^L}{\arg\min} \left(E_r \right) \tag{3.2}$$

– Para contar con el efecto de la deformación presente en el submapa local, modificamos el BA incluyendo un cálculo mecánico de la misma. Se construye un modelo de elementos finitos (FEM) con los puntos 3D del submapa local, y se calcula la energía de deformación causada por el BA al modificar sus coordenadas.

3.3. Modelo FEM del mapa local

Para calcular la energía de deformación fruto del desplazamiento en los puntos 3D del submapa local, estos se emplean para construir un modelo de elementos finitos.

Mallado de la nube de puntos del submapa local

1. De entre todos los puntos 3D presentes en el submapa local, se toma un subconjunto compuesto por aquellos que tienen un correspondiente en el mapa global, ejemplificado en la Figura 3.5a.

2. Se le aplica una aproximación por Moving-Least-Squares (MLS), que suaviza la nube de puntos aproximando la superficie a un polinomio y elimina espurios.

3. Los puntos se proyectan sobre una superficie plana y se lleva a cabo una triangulación de Delaunay que proporciona una malla plana de elementos triangulares como la de la Figura 3.5b. Esta triangulación se emplea para construir elementos prisma triangular (C3D6) o hexaedros (C3D8):

   a) Si se ha escogido elemento C3D6, esta malla se replica a una distancia calculada en función de los tamaños de los elementos, que hace de altura para el elemento. Ambas mallas se conectan para obtener prismas triangulares como se muestra en las Figuras 3.5c y 3.5d.

   b) Si se trabaja con elementos C3D8, los triángulos se convierten en cuadriláteros, para ello, se añaden nodos adicionales en el punto medio de cada segmento que conforma los triángulos y en el ortocentro, obteniendo las coordenadas de cada nuevo punto en función de los nodos próximos, como se representa en la Figura 3.5e. Tras la transformación a cuadriláteros, se genera una malla análoga a una distancia pre-establecida y se conectan ambas capas para obtener los hexaedros.

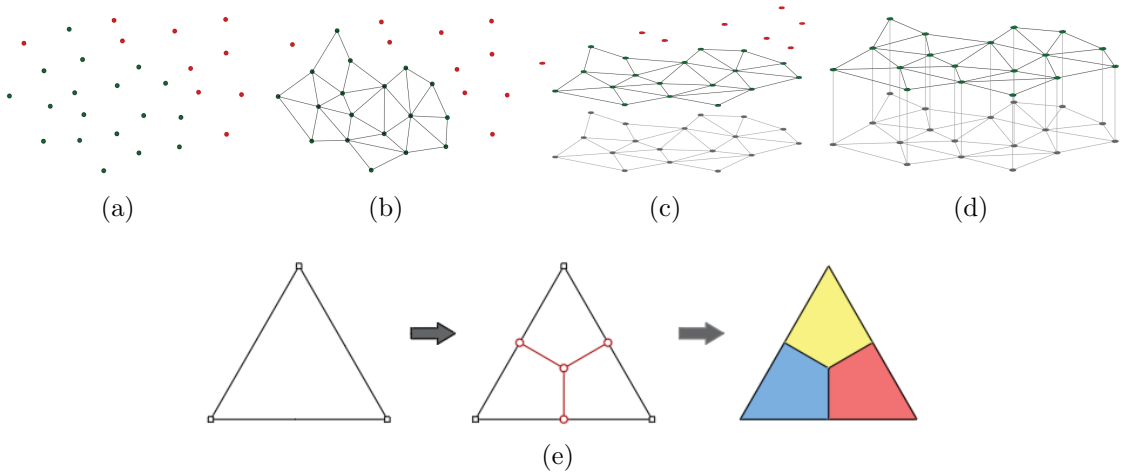


Figura 3.5: Etapas del proceso de mallado para prisma triangular. (a) Nube con puntos del submapa local donde • representa todos aquellos con un punto correspondiente en el mapa global y • aquellos sin correspondencia. (b) Triangulación de la nube. (c) Réplica de la malla a distancia predefinida. (d) Unión de ambas capas para formar los prismas triangulares. (e) Transformación de cada triángulo en 3 cuadriláteros.

Resolución por elementos finitos de la malla

1. Siguiendo los resultados de [11] seleccionamos módulo elástico $E = 5{,}18MPa$ y coeficiente de Poisson $\nu = 0{,}4999$ como parámetros de Lamé representativos del tejido. El espesor elemental h se ha calculado como la longitud media de todos los segmentos que definen todos los elementos.

2. Con la malla definida se construye una matriz de rigidez $\boldsymbol{K}$ que define su comportamiento; dado que los elementos no son regulares, el proceso de ensamblaje incluye el cálculo individual de la matriz de rigidez elemental $\boldsymbol{Ke}$ para cada elemento C3D8 y/o C3D6, cuya forma en coordenadas naturales se muestra en las Figuras 3.6a y 3.6b.

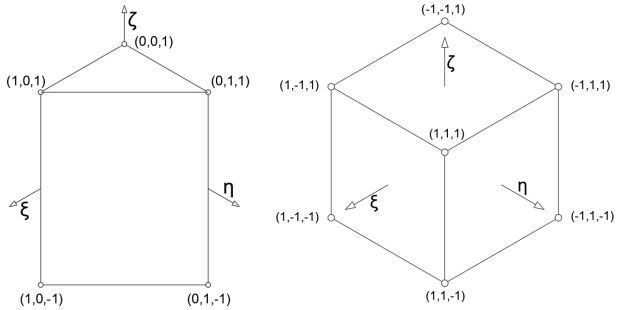
3. Se imponen condiciones de frontera de Dirichlet a la matriz de rigidez, estableciendo un encastre en la cara inferior.

4. Antes de iniciar la optimización no lineal se registra la posición inicial de los nodos que participarán en un vector u_0.

5. Tras cada paso de la optimización se registra la nueva posición u_f de los puntos presentes en la cara vista de la malla, manteniendo constante la posición de los puntos de la base, en condiciones de encastre. Con las posiciones finales se calcula el desplazamiento de los puntos respecto de la posición de partida y se imponen condiciones de Dirichlet al vector de desplazamientos correspondientes a las aplicadas sobre la matriz de rigidez.

6. Con desplazamientos y matriz de rigidez definidos puede obtenerse la fuerza sobre los nodos, que permite calcular la energía de deformación inducida en el modelo de acuerdo con la Ecuación 3.3.

$$E_{FEA} = a \cdot F = a' \cdot K \cdot a \tag{3.3}$$



(a) (b)

Figura 3.6: Elementos de referencia con aproximación lineal: (a) prisma triangular (C3D6) y (b) hexaedro (C3D8).

Función de coste

La nueva función de coste a minimizar en el BA se muestra en la Ecuación 3.4 y consta de dos componentes:

- Error de reproyección: obtenido como resultado de proyectar los puntos 3D comunes a ambos mapas, los del mapa global sobre las imágenes del local y viceversa. Durante la optimización, la posición 3D de los puntos del modelo global X_i^G permanece fija, modificando solo la correspondiente a los puntos del modelo local X_i^L; también se optimizan los parámetros extrínsecos de las imágenes del modelo local $\theta_{ext}^{j,L}$.

- Energía de deformación: mostrada en la Ecuación 3.3 y causada por el desplazamiento impuesto a los puntos del modelo local X_i^L respecto de su posición original.

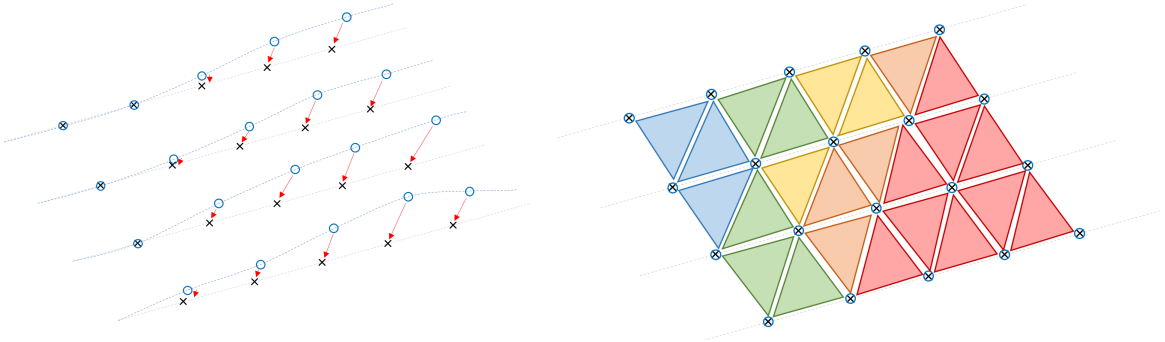
A mayor desplazamiento de los puntos, más se reduce el coste asociado al error de reproyección, pero aumenta el debido a la deformación. El algoritmo busca una posición de equilibrio entre ambos factores modificando la posición de los puntos 3D y poses de cámara del submapa local. Se aplica un peso w_{FEA} al valor de energía de deformación para hacer comparables ambos factores. El peso se ha calculado empíricamente y es diferente para la iteración inicial y las sucesivas.

$$\underset{\mathbf{X}_i^L, \theta_{ext}^L}{\arg\min} \left(E_r + w_{FEA} \cdot E_{FEA} \right) \tag{3.4}$$

3.4. Estimación de deformaciones

Con ambos mapas alineados se puede medir la diferencia entre estos. Mediante el siguiente procedimiento presentamos un valor cuantitativo en forma de Energía de Deformación Elástica necesaria para deformar el submapa local hasta que este ocupa la posición del global:

1. Alineación de submapa local respecto del mapa global.
2. Siendo conocidos los puntos 3D comunes a ambos mapas, imposición de un desplazamiento a los puntos del submapa local de modo que para cada uno de ellos se alcance $X_i^L = X_i^G$ como representado en la Figura 3.7a.
3. Se mide la energía de deformación resultado de este desplazamiento, obteniendo una discretización de la misma por elemento como se indica en la Figura 3.7b



(a) (b)

Figura 3.7: (a) Se representan con $\circ$ los puntos del submapa local y con $\times$ los puntos del global. Se impone un desplazamiento $\leftarrow$ de los primeros hasta que ocupen la misma posición 3D que los segundos. (b) Muestra la distribución de energía de deformación por elemento, resultado de imponer los desplazamientos anteriores.

Capítulo 4

Validación experimental

La validación del proyecto se ha realizado empleando secuencias de endoscopia médica *in vivo* grabadas como parte del proyecto Endomapper [12]. El análisis de resultados se ha obtenido a partir de la secuencia *Seq_001* del dataset, que muestra una colonoscopia completa de la que se han extraído dos subconjuntos de imágenes en las que el endoscopio recorre el colon transverso.

Los dos subconjuntos de imágenes, que se indican en la Tabla 4.1, se combinan para formar una única secuencia de vídeo. En la Figura 4.1 se presentan imágenes representativas de la porción de secuencia empleada. Las imágenes se han seleccionado de modo que faciliten el proceso del SfM, esto es, que no presenten oclusiones importantes causadas por el propio tejido o suciedad, con ausencia de líquido, reflejos y movimientos bruscos del endoscopio.

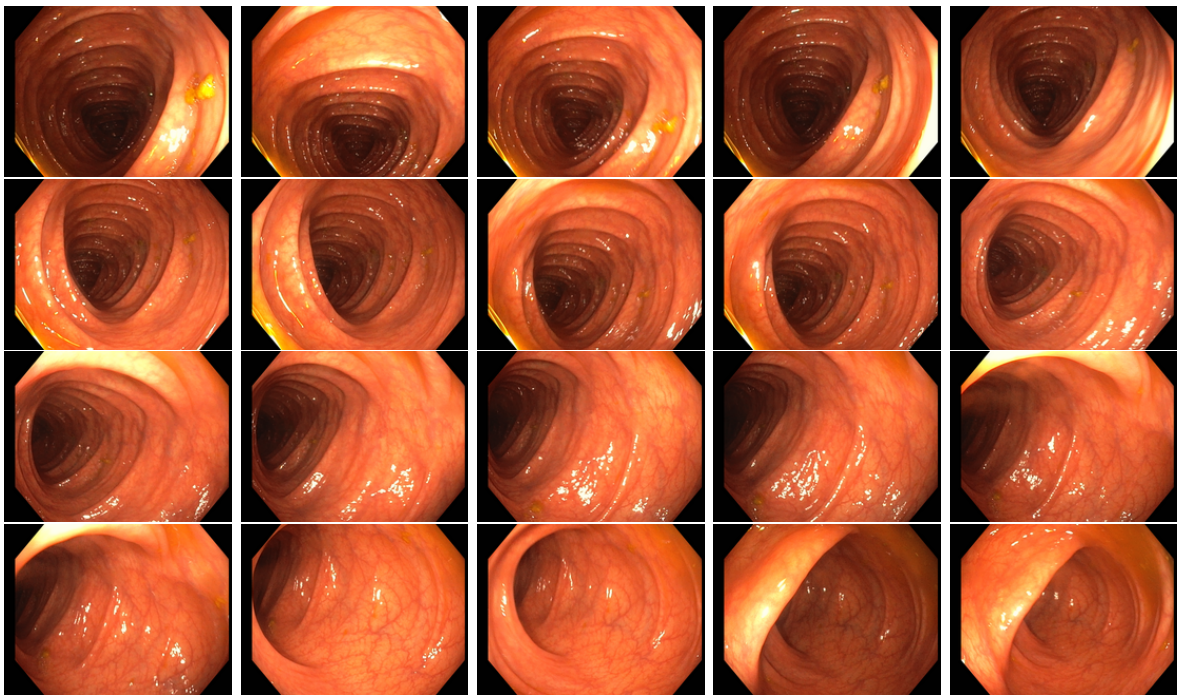


Figura 4.1: Imágenes de la porción de secuencia empleada en la validación, en la que el endoscopio retrocede por el colon transverso.

El análisis se ha realizado sobre CPU en un equipo con procesador Intel(R) Core(TM) i7-12700H a 2.30 GHz y 32 Gb de memoria RAM DDR5 a 4800 MHz.

Sub-sequencia	Inicio		Fin		Total	
	Imagen	Tiempo	Imagen	Tiempo	Imágenes	Tiempo
1	9410	3:55.250	9600	4:00.250	192	0:05.000
2	9700	4:02.500	9960	4:09.000	262	0:06.500
$1 \cup 2$	-	-	-	-	454	0:11.500

Tabla 4.1: Tramos de secuencia empleados en la validación. Los dos subconjuntos de imágenes se combinan en un único vídeo de 454 imágenes. Tiempos en [mm:ss.ms].

4.1. Geometría epipolar en cámara FishEye

El nuevo método de búsqueda guiada de correspondencias y verificación geométrica expuesto en el Capítulo 2 se aplica a la secuencia de estudio obteniendo los resultados mostrados en la Tabla 4.2:

- Durante el proceso de búsqueda de correspondencias entre imágenes, el algoritmo desarrollado obtiene aproximadamente el doble de resultados que el original.
- Procesando la misma secuencia de imágenes, el algoritmo de referencia genera dos reconstrucciones de 245 y 162 imágenes cada una, mientras que el nuevo método reconstruye el total de la escena en un único mapa empleando 363 imágenes.
- El número de puntos triangulados con nuestro proceso es un 50 % inferior respecto del original, sin embargo, la media de observaciones por punto es significativamente superior, indicando una mayor precisión en la triangulación de los puntos obtenidos, al estar realizada respecto de más imágenes.

	v0	v1
Número de mapas globales	2	1
Longitud track 1 (número de imágenes mapa)	245	363
Longitud track 2 (número de imágenes mapa)	162	-
Puntos por imagen (media)	313.7	639.8
Número total de correspondencias entre imágenes	141788	289190
Número total de puntos 3D (triangulados)	34285	17128
Observaciones por punto 3D (media)	6.03	7.56

Tabla 4.2: Efecto de actualizar la búsqueda guiada de correspondencias de acuerdo a la geometría de la cámara FishEye. v0 corresponde a los resultados obtenidos con un SfM de referencia, v1 presenta resultados obtenidos con el nuevo algoritmo.

4.2. Efecto de la compresión con pérdidas en la precisión del mapa

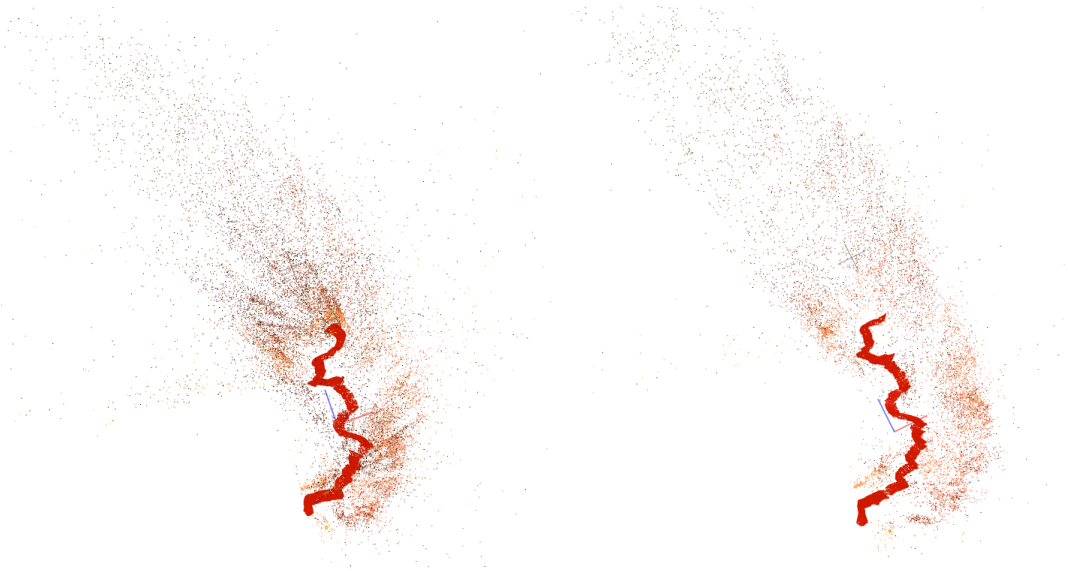
Las imágenes obtenidas en el procedimiento médico son de alta resolución y tamaño, requiriendo un alto coste computacional para ser procesadas; por ello, es práctica habitual aplicar una compresión a las imágenes. A título orientativo, en la secuencia de estudio (completa), esta compresión permite reducir el tamaño original de 12.6 Gb para un vídeo de 6 minutos y 10 segundos, a 356.7 Mb. Si bien este proceso no afecta a la resolución de las imágenes, sí modifica la codificación de los archivos, lo cual puede afectar negativamente al funcionamiento del SfM.

En la Tabla 4.3 se presenta el efecto de aplicar una compresión en las imágenes. Aquellas sin preprocesar permiten detectar un 62 % más de puntos característicos, dando lugar a un 13 % extra de correspondencias. Se encuentra que, durante el proceso de construcción del mapa, el número de puntos triangulados en imágenes sin pérdidas es aproximadamente un 50 % inferior al obtenido empleando imágenes con pérdidas; sin embargo, en la Figura 4.2 se muestran los modelos obtenidos tras generar un modelo con y sin pérdidas y puede observarse cómo la dispersión del modelo sin pérdidas es significativamente menor y con los puntos distribuidos representando la forma cilíndrica de la cavidad intestinal; lo cual unido a un mayor número de observaciones por punto indica mayor calidad de la reconstrucción.

	Cantidad [-]		Tiempo [min]	
Compresión con pérdidas	Si	No	Si	No
Puntos de imagen detectados	535184	**868223**	**0.747**	0.813
Correspondencias entre imágenes	2107776	**2391086**	**14.936**	24.461
Imágenes añadidas al mapa	452	**363**	12.069	**8.679**
Puntos triangulados	34538	**17128**		
Observaciones por punto (media)	6.07	7.49		
Error de reproyección medio por punto	2.54	1.45		
Total	-	-	**27.752**	33.953

Tabla 4.3: Efecto de preprocesar las secuencias añadiendo una compresión con pérdidas en las imágenes. Se presenta el impacto en las etapas del SfM: detección de puntos característicos en las imágenes, emparejamiento de puntos entre imágenes, construcción del mapa (imágenes empleadas, puntos 3D generados, número de observaciones de cada punto 3D). Los valores más favorables se resaltan en negrita.

Como punto negativo, el tiempo necesario para procesar imágenes sin pérdidas aumenta un 22 %, siendo necesarios aproximadamente 34 minutos para construir el mapa, frente a los casi 28 minutos del proceso con pérdidas. Mención a que el tiempo requerido para la etapa de construcción del mapa es inferior en imágenes sin pérdidas debido al menor número de puntos y mayor número de observaciones, que resulta en una reconstrucción más precisa que requiere menos etapas de BA para su refinado.



(a) (b)

Figura 4.2: Modelo global obtenido empleando imágenes con pérdidas (a) y sin pérdidas (b). Los iconos rojos que se encuentran en el centro de las figuras representan las posiciones que se han calculado para cada imagen, en su conjunto indican la trayectoria seguida por el endoscopio. En torno a estas podemos ver los puntos 3D.

4.3. Alineamiento mapa global-submapa local

Empleando las modificaciones anteriores se procesa la secuencia obteniendo un mapa global que incluye 363 imágenes y 17128 puntos 3D, así como 49 submapas locales, cada uno de ellos aproximadamente 20 imágenes y alrededor de 1000 puntos que se indican en el histograma de la Figura 4.3a. Fijando en el espacio el mapa global, cada uno de los submapas locales se alinea respecto del primero de acuerdo con lo presentado en la Sección 3.2. Para cada alineamiento se han anotado valores de número de puntos y frames comunes entre el mapa global y el submapa local, que se presentan en la Figura 4.3b:

- Los submapas 25 y 28, con menor número de puntos y de imágenes en el segundo caso, corresponden con conjuntos de imágenes de peor calidad, en cuanto a que se encuentran próximas a la pared intestinal, siendo menos aprovechables.

- Las imágenes empleadas para construir los submapas 12 y 27 no se encuentran entre las empleadas para construir el mapa global, con lo que estos no pueden ser empleados en la comparación.

Así mismo, se registran valores de error de reproyección y energía de deformación obtenidos tras completar el proceso de alineamiento, que se presentan en los histogramas de las Figuras 4.4a y 4.4b, respectivamente.

- Se observa como el **error de proyección** es comparable entre todos los submapas, con la única excepción del número 26. De acuerdo a la Figura 4.3b, este

modelo está formado por un número bajo de puntos en comparación con el resto, revisando las estadísticas del SfM, se tiene que también ha obtenido un número menor de observaciones para cada punto 3D, lo que pudiera haber llevado a una triangulación menos precisa, explicando este mayor error de reproyección.

– Contrariamente al caso anterior, en la **energía de deformación** hay una gran dispersión en los resultados, tema que se abordará en la próxima sección.

4.4. Estimación de deformaciones

Por último, presentamos detalle de dos de los submapas locales, que de acuerdo a la sección anterior tienen características diferentes:

– Submapa 17: número elevado de puntos comunes con el mapa global y **baja** energía de deformación.

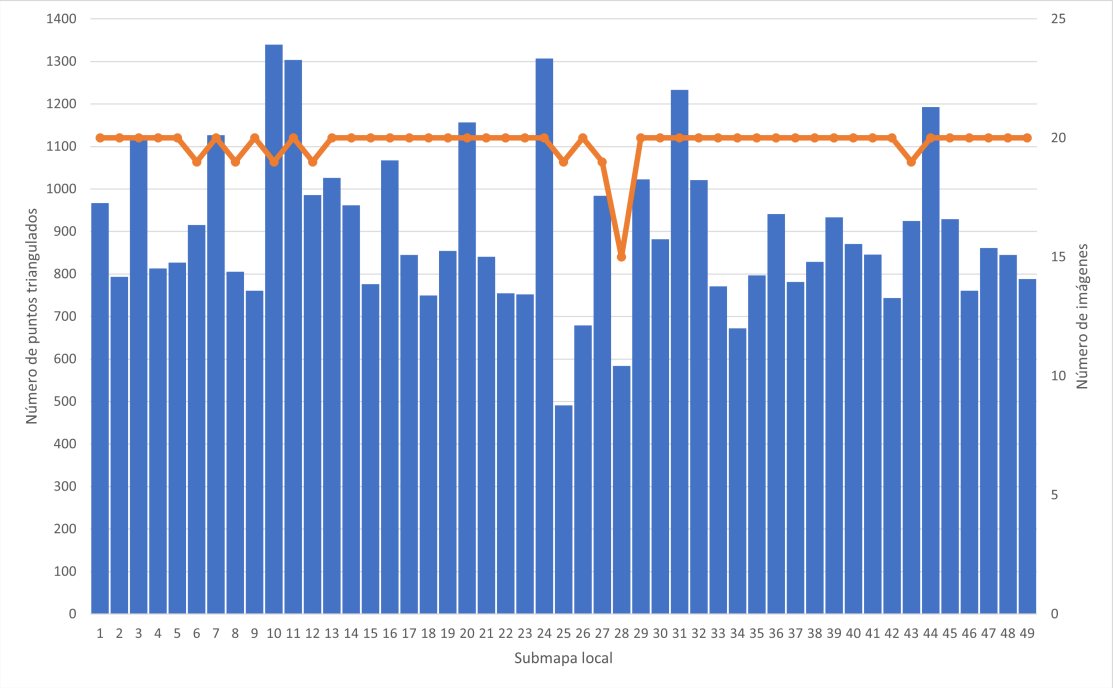
– Submapa 39: número medio de puntos comunes y **elevada** energía de deformación.

Comenzando con el submapa 17 para el que se ha medido baja energía, se construye el modelo de elementos finitos mostrado en la Figura 4.6, siendo este una malla compleja que abarca gran parte de la escena.

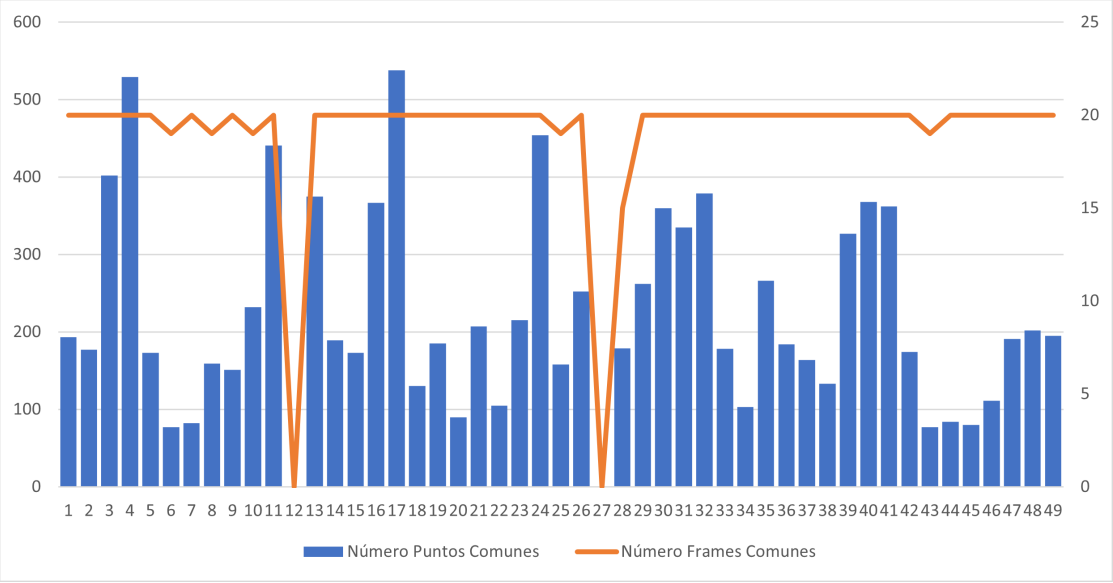
En cuanto al submapa 39, parte de las imágenes que lo componen se presentan en la Figura 4.5 y sobre ellas se ha indicado el área de tejido con mayor deformación, siendo esta apreciable a simple vista.

De acuerdo a lo expuesto en la Sección 3.4, los puntos 3D de este submapa se emplean para construir modelos de elementos finitos sobre los que se imponen desplazamientos hasta que su configuración sea la misma que la del mapa global para el mismo entorno, obteniendo la distribución de energía elemental mostrada en la Figura 4.7a, en la que se observa como aquellos elementos próximos a la región de alta deformación presentan tonos cálidos indicando la presencia de esta.

Realizando el mismo procedimiento para el modelo de baja deformación se obtiene la distribución mostrada en la Figura 4.7b, donde además de una malla más elaborada correspondiente al mayor número de puntos, esta se ha coloreado automáticamente en todos fríos correspondientes a poca deformación.

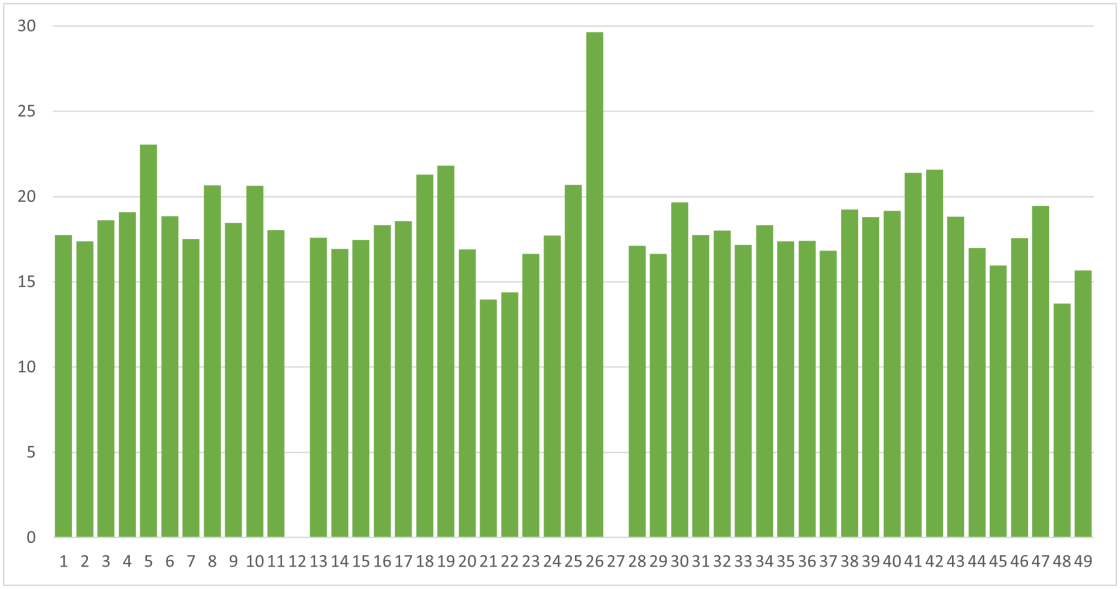


(a)

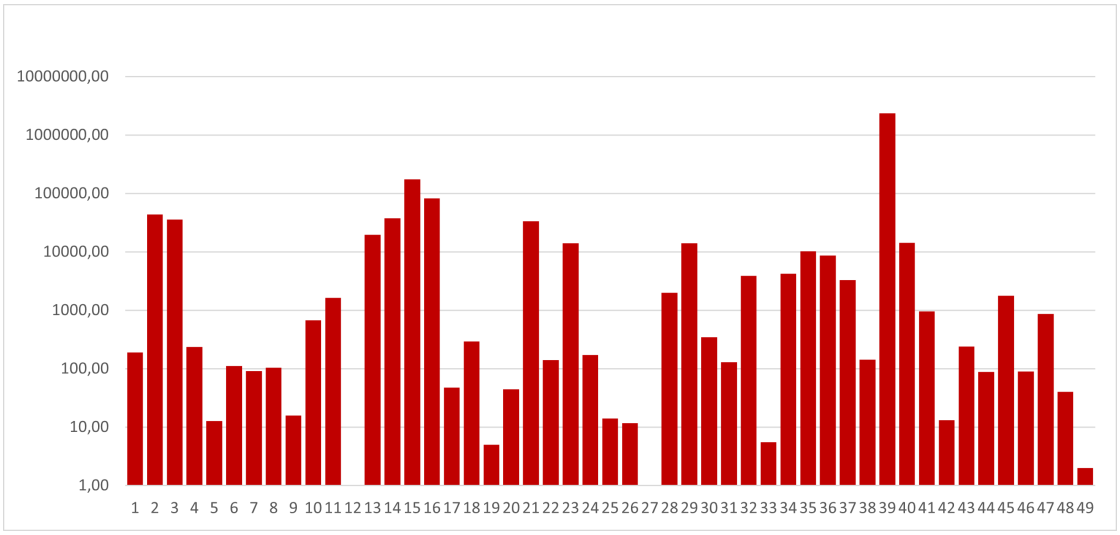


(b)

Figura 4.3: (a) Distribución de puntos e imágenes que forman parte de cada submapa local. Los bin azules, alineados con el eje izquierdo, representan el número de puntos por modelo, la línea naranja, alineada con el eje derecho, indica el número de imágenes. (b) Alineados con el eje izquierdo, los bins azules indican el número de puntos comunes entre el mapa global y cada submapa local. Alineada con el eje derecho, la línea naranja indica el número de frames comunes a ambos modelos.



(a)



(b)

Figura 4.4: (a) Error de reproyección obtenido tras completar la alineación expresado en píxeles. (b) Energía de deformación resultado de imponer movimientos en los puntos del submapa local durante el proceso de alineamiento, expresada en píxeles y con el eje izquierdo en escala logarítmica en base 10.

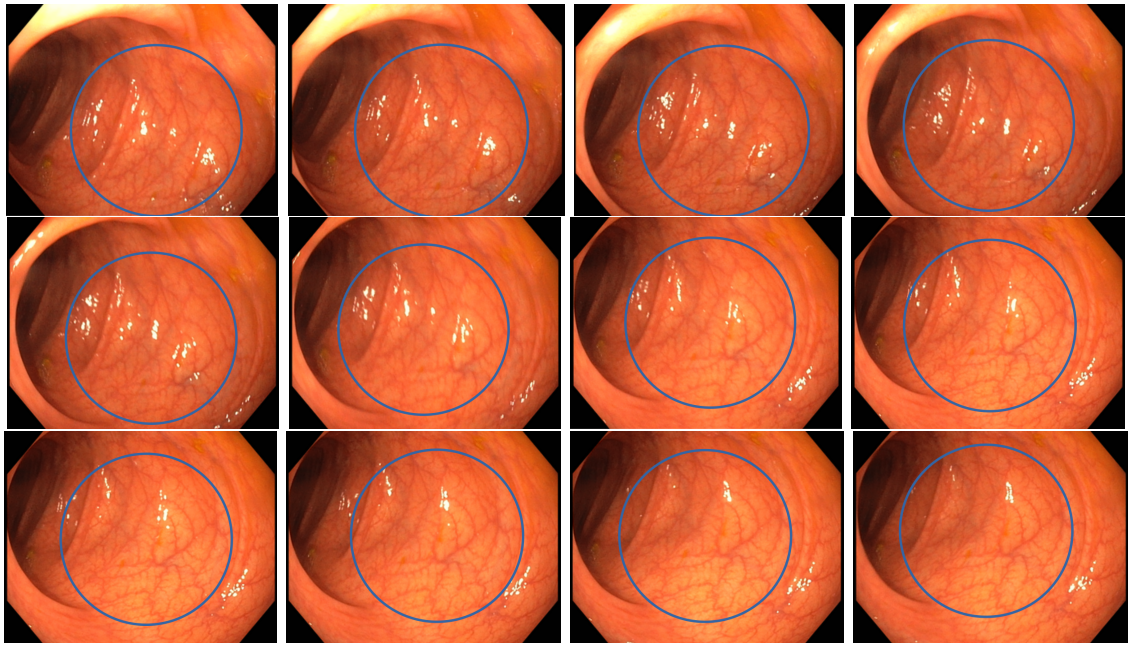
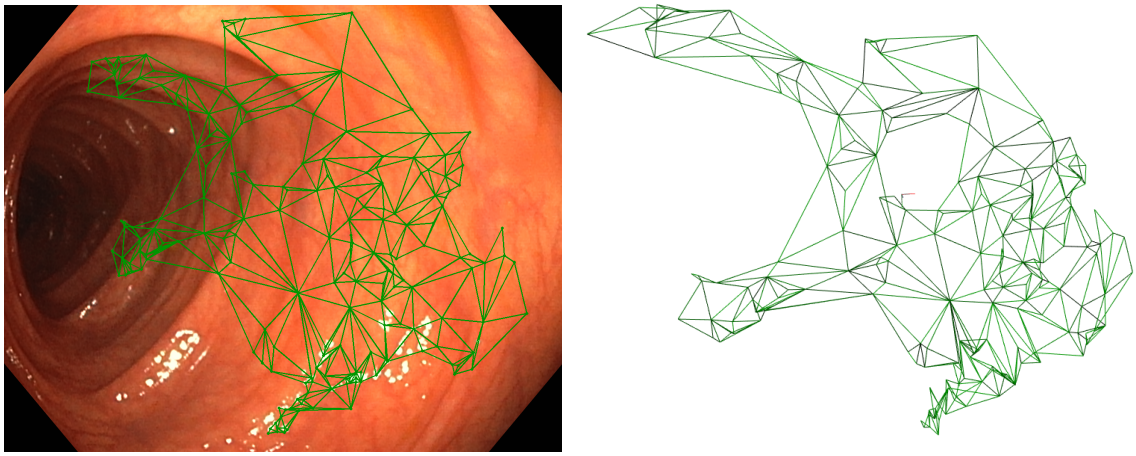
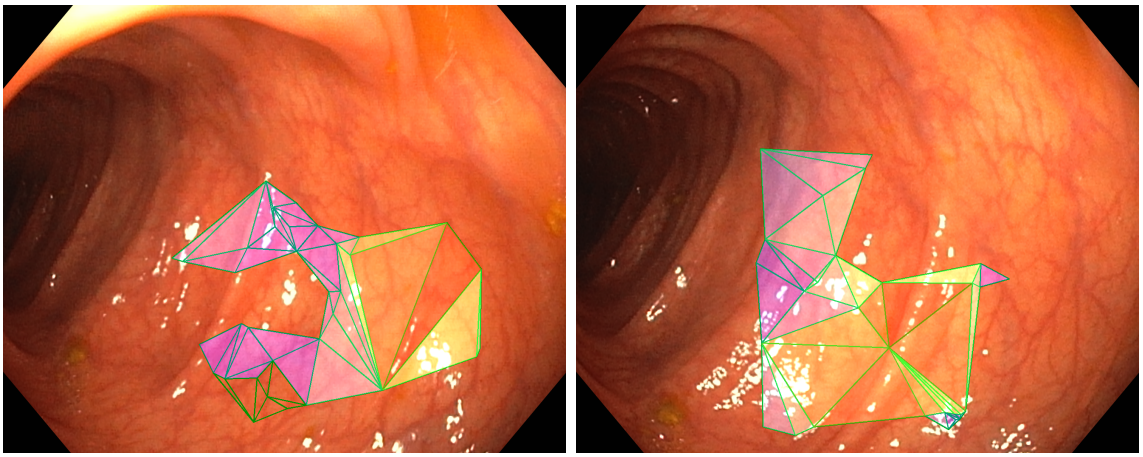


Figura 4.5: Imágenes de tejido con deformación visible, con la zona de interés indicada en azul.

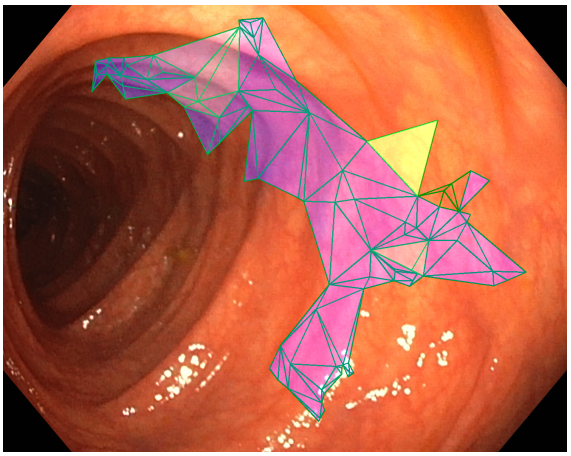


(a) (b)

Figura 4.6: (a) Malla generada con el submapa local 17 y dibujada sobre la imagen escogida como anclaje. (b) Visualización 3D de la misma malla.



(a)



(b)

Figura 4.7: (a,b) Distribución de energía sobre parte de la malla obtenida con el modelo de elevada energía de deformación. (c) Distribución de energía sobre una malla generada en un modelo con baja energía de deformación.

Capítulo 5

Resultados

5.1. Conclusiones y discusión

Se ha estudiado la viabilidad del procesado de secuencias de endoscopia con algoritmos de Structure from Motion, implementando modificaciones que mejoran las reconstrucciones obtenidas: un nuevo método de búsqueda guiada de correspondencias y verificación geométrica optimizado para las cámaras fish-eye que portan los endoscopios; obteniendo una considerable mejora de proceso con más puntos triangulados.

Se ha estudiado el efecto de aplicar compresión con pérdidas a las secuencias de endoscopia, concluyendo que trabajar con imágenes sin compresión es favorable para obtener reconstrucciones de mayor calidad, permitiendo obtener hasta un 60 % más de información de las escenas. Si bien el tiempo de cálculo es mayor en las etapas de detección de puntos de imagen y emparejamiento, este disminuye en la construcción del mapa, consiguiendo un menor número de puntos pero con mas observaciones, es decir, un mapa más preciso.

Para identificar la presencia de deformación en el tejido, se genera un mapa que contiene la envolvente rígida de una escena y se compara con mapas de pequeño tamaño que capturan tejido deformado. Los mapas se alinean entre sí mediante una optimización no lineal que minimiza conjuntamente el error de reproyección de los puntos triangulados y la energía de deformación de la escena.

Para lograr lo anterior se ha implementado un método de análisis por elementos finitos, que proporciona la distribución de energía de deformación en una porción de tejido.

5.2. Trabajo futuro

El trabajo realizado permite emplear herramientas de análisis de sólidos deformables en software de SfM. Se propone profundizar en el cálculo mecánico por las siguientes vías:

- Evaluar nuevos tipos de elemento en el cálculo por elementos finitos.

- Se ha encontrado que con frecuencia el mallado presenta irregularidades debido a la posición errática de los puntos así como a la estructura de la escena; un método de mallado específicamente diseñado para escenas cilíndricas (semejantes al intestino) podría mejorar el resultado.

- El análisis se ha realizado fijando los parámetros mecánicos (E, ν) en base a las referencias encontradas en la literatura; sin embargo, se ha encontrado que los valores difieren en gran medida entre diferentes fuentes. Se propone tomar mediciones de la deformación real existente y utilizar estos datos para inferir los parámetros.

- Entrenamiento de modelos de aprendizaje automático empleando la información medida como etiquetas.

- El método de alineamiento de nubes de puntos puede modificarse para su integración en el proceso de construcción de mapas del SfM, avanzando hacia una implementación completa de un SfM para entornos deformables.

Capítulo 6

Bibliografía

[1] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.

[2] Robert C Bolles, H Harlyn Baker, and David H Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International journal of computer vision*, 1(1):7–55, 1987.

[3] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[4] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[6] Tony Lindeberg. *Scale Invariant Feature Transform*, volume 7, chapter 7. Scholarpedia, 05 2012.

[7] Martin A Fischler and Robert 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.

[8] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Workshop on Vision Algorithms*, 1999.

[9] Juho Kannala and Sami Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28:1335–40, 09 2006.

[10] Alain Pagani and Didier Stricker. Structure from motion using full spherical panoramic cameras. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 375–382, 2011.

[11] Ben Christensen, Kevin Oberg, and Jeffrey Wolchok. Tensile properties of the rectal and sigmoid colon: A comparative analysis of human and porcine tissue. *SpringerPlus*, 4, 03 2015.

[12] Pablo Azagra, Carlos Sostres, Ángel Ferrandez, Luis Riazuelo, Clara Tomasini, Oscar León Barbed, Javier Morlana, David Recasens, Victor M Batlle, Juan J Gómez-Rodríguez, et al. Endomapper dataset of complete calibrated endoscopy procedures. *arXiv preprint arXiv:2204.14240*, 2022.

[13] Íñigo Cirauqui. Point cloud comparer, howpublished = "`https://github.com/icirauqui/pc_compare`", year = 2022.

[14] Íñigo Cirauqui. Finite element analisys. `https://github.com/icirauqui/FEA2`, 2020.

[15] Íñigo Cirauqui. Colmap for colonoscopy 2, howpublished = "`https://github.com/icirauqui/colmap_for_colonoscopy_2`", year = 2022.

[16] Íñigo Cirauqui. Point cloud comparer, howpublished = "`https://github.com/icirauqui/pc_compare`", year = 2022.

[17] Rao Garimella and Rao Garimella. Title: A Simple Introduction to Moving Least Squares and Local Regression Estimation Intended for: A Simple Introduction to Moving Least Squares and Local Regression Estimation. Technical report, Los Alamos National Laboratory, 06 2017.

[18] Dassault Systèmes. Abaqus cae portal. `https://www.3ds.com/products-services/simulia/products/abaqus/abaquscae/`, 2020.

Anexo

Anexo A

Software desarrollado

Todo el software desarrollado se encuentra disponible en GitHub:

- **Cálculo del error angular:** disponible en [13].
- **Análisis por elementos finitos:** disponible en [14].
- **Versión modificada del software de SfM COLMAP:** disponible en [15].
- **Alineación de nubes de puntos:** disponible en [16].

A continuación se presenta detalle de la clase desarrollada para el análisis por elementos finitos. El resto de proyectos contarán con documentación referenciada directamente en los repositorios.

A.1. Desarrollo y validación del análisis por elementos finitos

Se ha desarrollado una clase para el análisis por elementos finitos, que se ha integrado en el proceso de optimización no lineal. El código cuenta con 3 bloques diferenciados de funciones. A continuación, se proporciona una descripción corta de cada una y un detalle de las funciones de cálculo de matriz elemental y ensamblaje, siendo estas las más relevantes para el trabajo.

1. Funciones generales: constructor, destructor, visualización de datos, guardado en disco y operaciones matemáticas.
    a) **Constructor:** la clase se genera con un identificador que la asocia al frame en que va a actuar. Además, se asignan propiedades del material y se calculan parámetros de Lamé y matriz de comportamiento. También se define el tipo de elemento a utilizar.
    b) **Destructor:** destructor vacío.
    c) **InvertMatrixEigen:** calcula la inversa de la matriz proporcionada.

d) **MultiplyMatricesEigen:** multiplica las matrices proporcionadas.

2. Funciones de mallado: carga y gestión de puntos, configuración de objetos, suavizado por *moving least squares* y mallado.

   a) **GetMapPointCoordinates:** extrae las coordenadas de los puntos del mapa con los que se trabajará. El vector de puntos ha sido llenado desde la función de relocalización.

   b) **LoadMPsIntoCloud:** genera el objeto nube de puntos de la librería PCL y carga la información de posición 3D y normales desde los puntos del mapa.

   c) **MLS:** aplica una aproximación por Moving Least Squares [17] que suaviza la malla, aproximándola a un polinomio, y elimina espurios.

   d) **Compute Mesh:** utiliza la nube suavizada para generar una malla triangular empleando la función GreedyProjection de la librería PCL.

   e) **CalculateGP3Parameters:** calcula los parámetros por los que se rige el proceso de mallado. El cálculo se lleva a cabo en función de la estructura de la nube de puntos.

   f) **tri2quad:** genera cuadriláteros a partir de una malla de triángulos.

   g) **SetSecondLayer:** replica la malla de triángulos a cuadriláteros a una distancia predeterminada para formar los elementos tridimensionales.

3. Funciones de análisis: cálculos de matrices de rigidez elementales, ensamblaje, calculo de desplazamientos, fuerzas y energías.

   a) **Set_u0:** carga la posición inicial de los puntos al comienzo de la optimización no lineal, se empleará como base para el cálculo de desplazamientos que darán la energía de deformación.

   b) **ComputeKeiC3D6:** calcula la matriz elemental para un prisma triangular en función de la posición de los nodos proporcionada. Se proporciona información teórica adicional en el Anexo 2.

   c) **ComputeKeiC3D8:** calcula la matriz elemental para un hexaedro en función de la posición de los nodos proporcionada. Se proporciona información teórica adicional en el Anexo 2.

   d) **MatrixAssemblyC3D6:** proceso de ensamblaje de matriz de rigidez de elementos prisma triangular. Se proporciona información teórica adicional en el Anexo 2.

   e) **MatrixAssemblyC3D8:** proceso de ensamblaje de matriz de rigidez de elementos hexaedro. Se proporciona información teórica adicional en el Anexo 2.

f) **ImposeDirichletEncastre_K:** impone las condiciones de contorno de Dirichlet a la matriz de rigidez previamente calculada. Se encastra la cara duplicada del modelo.

g) **ImposeDirichletEncastre_a:** impone las condiciones de contorno de Dirichlet al vector de desplazamientos previamente calculado.

h) **Set_uf:** carga las posiciones de los puntos actualizadas en cada etapa de la optimización por Levenberg Marquardt.

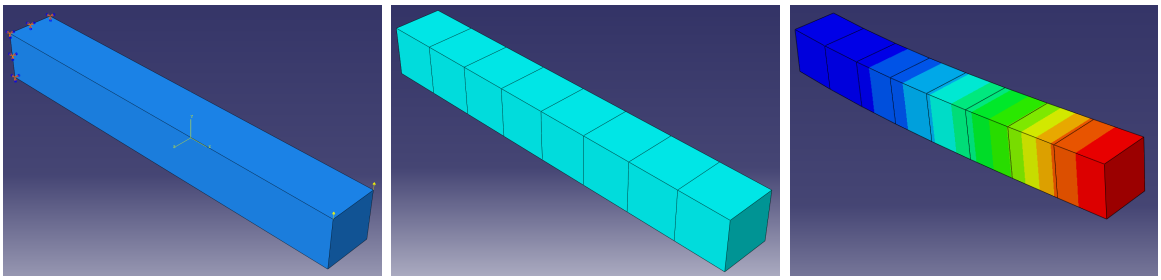
i) **ComputeDisplacement:** calcula el desplazamiento de los puntos tras el establecimiento de las posiciones finales.

j) **ComputeForces:** calcula la fuerza necesaria para los desplazamientos generados anteriormente.

k) **ComputeStrainEnergy:** calcula la energía de deformación elástica causada por esos desplazamientos.

l) **NormalizeStrainEnergy:** normaliza la energía de deformación calculada.

Se presenta una verificación contra el software Abaqus CAE [18]. Para ello, se analiza una viga simple de perfil cuadrado como la de la Figura A.1a de dimensiones 8x1x1, esta se encastra en el extremo izquierdo y se aplican dos fuerzas puntuales de 0.5 N en los nodos superiores del extremo derecho. Se genera un mallado en hexaedros C3D8 de lado 0.25, dando lugar a los 64 elementos mostrados en la Figura A.1b. Se resuelve el modelo dando lugar a la deformación mostrada en la Figura A.1c.



(a) (b) (c)

Figura A.1: Modelo en Abaqus con elementos C3D8. (a) Muestra una viga de perfil cuadrado con encastre en un extremo y fuerzas puntuales en el contrario. (b) Mallado aplicado. (c) Deformación generada.

Se obtienen, para los nodos de la cara en la que se aplican las fuerzas, los desplazamientos de la Tabla A.1. La configuración de nodos, encastre y fuerzas se carga a la versión de la librería disponible en [14], los mismos archivos empleados para este análisis se incluyen en la carpeta *input_C3D8* a modo de ejemplo. Se ejecuta el programa dando lugar a los desplazamientos del segundo bloque de columnas de la Tabla A.1.

Nodo	Abaqus			FEA2			Error(%)		
Nodo	u_x	u_y	u_z	u_x	u_y	u_z	e_x	e_y	e_z
1	-0.021	0.198	0	-0.021	0.198	0	0	0	0
2	-0.021	0.198	0	-0.021	0.198	0	0	0	0
3	-0.02	0.156	0.001	-0.02	0.156	0.001	0	0	0
4	-0.02	0.156	-0.001	-0.02	0.156	-0.001	0	0	0
5	-0.018	0.116	0.001	-0.018	0.117	0.001	0	1	0
6	-0.018	0.116	-0.001	-0.018	0.116	-0.001	0	0	0
7	-0.016	0.081	0.001	-0.016	0.081	0.001	0	0	0
8	-0.016	0.081	-0.001	-0.016	0.081	-0.001	0	0	0
9	-0.013	0.051	0.002	-0.013	0.051	0.002	0	0	0
10	-0.013	0.051	-0.002	-0.013	0.051	-0.002	0	0	0
11	-0.01	0.027	0.002	-0.01	0.027	0.002	0	0	0
12	-0.01	0.027	-0.002	-0.01	0.027	-0.002	0	0	0
13	-0.006	0.011	0.002	-0.006	0.011	0.002	0	0	0
14	-0.006	0.011	-0.002	-0.006	0.011	-0.002	0	0	0
15	-0.002	0.003	0.002	-0.002	0.003	0.002	0	0	0
16	-0.002	0.003	-0.002	-0.002	0.003	-0.002	0	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0.021	0.197	-0.001	0.021	0.197	-0.001	0	0	0
20	0.021	0.197	0.001	0.021	0.197	0.001	0	0	0
21	0.02	0.156	-0.001	0.02	0.156	-0.001	0	0	0
22	0.02	0.156	0.001	0.02	0.156	0.001	0	0	0
23	0.018	0.116	-0.001	0.018	0.117	-0.001	0	1	0
24	0.018	0.116	0.001	0.018	0.116	0.001	0	0	0
25	0.016	0.081	-0.001	0.016	0.081	-0.001	0	0	0
26	0.016	0.081	0.001	0.016	0.081	0.001	0	0	0
27	0.013	0.051	-0.002	0.013	0.051	-0.002	0	0	0
28	0.013	0.051	0.002	0.013	0.051	0.002	0	0	0
29	0.01	0.027	-0.002	0.01	0.027	-0.002	0	0	0
30	0.01	0.027	0.002	0.01	0.027	0.002	0	0	0
31	0.006	0.011	-0.002	0.006	0.011	-0.002	0	0	0
32	0.006	0.011	0.002	0.006	0.011	0.002	0	0	0
33	0.002	0.003	-0.002	0.002	0.003	-0.002	0	0	0
34	0.002	0.003	0.002	0.002	0.003	0.002	0	0	0
35	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0

Tabla A.1: Comparativa de resultados Abaqus-C++ en metros y % de error para una precisión de milímetros.

Lista de Figuras

Lista de Tablas