Universidad Zaragoza
1542

Trabajo Fin de Grado en Ingeniería Industrial

Tracking, deformación y localización en secuencias de endoscopia médica

Autor

Blanca Guillén Cebrián

Directores

José María Martínez Montiel

Escuela de Ingeniería y Arquitectura

2018

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Blanca Guillén Cebrián,

con nº de DNI 25207680W 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 (Grado/Máster) Grado, (Título del Trabajo) Tracking, deformación y localización en secuencias de endoscopia médica

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

Zaragoza, 31 de agosto de 2018

Fdo: _____

Resumen

Partiendo de un sistema ORB-SLAM monocular deformable, que inicialmente operaba sobre un prototipo de laboratorio compuesto de una tela que sufre deformaciones, se ha adaptado para que procese por primera vez secuencias médicas de endoscopias reales in-vivo de un dataset estandarizado. En las secuencias, el endoscopio se mueve y la escena se deforma. El objetivo es estimar, para cada frame del vídeo, la forma 3D de la superficie y la posición del endoscopio relativa a esa superficie.

El sistema inicialmente realiza una exploración de la escena en reposo mediante un sistema de ORB-SLAM rígido, creando un mapa de puntos en 3D a partir del cual se estima, mediante el método de Poisson, la superficie en reposo de la escena. Esta superficie en reposo se denomina plantilla y se codifica mediante una malla triangular.

El sistema inicial tenía dos obstáculos para procesar las imágenes médicas: 1) imposibilidad de crear la malla en escenas médicas debido a los puntos espurios en la estimación del mapa, 2) baja tasa de emparejamientos debido a la falta de repetitividad de los puntos ORB. Los dos obstáculos han sido superados. El primero mediante la inclusión de un algoritmo de LMedS (Least Median of Squares) para la detección del rango válido de profundidades. El segundo mediante la inclusión de una nueva etapa de búsqueda por correlación para emparejar los puntos del mapa que no han sido emparejados mediante los puntos ORB.

El sistema ha sido validado experimentalmente en tres secuencias médicas con deformación. Para ello, la superficie ground truth se estima a partir de una segunda cámara estéreo. El algoritmo propuesto consigue triplicar la tasa de emparejamiento hasta más del 95 %, a la vez que se mantiene el error en la estimación de la superficie deformable y con un costo computacional en torno a los 10 ms.

La implementación se ha hecho en C++ y está disponible en un repositorio privado de GitHub.

Agradecimientos

Agradecimiento a la beca PEX-17-040 de iniciación a la investigación financiada por la línea de transferencia LTI3A0105 de la Universidad de Zaragoza.

Agradezco a José Lamarca por la asistencia y ayuda en la comprensión de su código. A Javier Morlana, por permitirme el uso de su código para calcular el ground truth del estéreo. A Ignacio Cuiral, por la ayuda en la obtención de datos de calibración en el Hamlyn Dataset.

Índice

1. Introducción y objetivos **4**
1.1. Introducción 4
1.2. Objetivos 5
1.3. Estructura del documento 5

2. SLAM Monocular en escenas deformables **7**
2.1. Modelado de la malla 7
2.2. Optimización de la deformación de la malla y posición de la cámara . . 9

3. Estimación de la plantilla para escenas médicas **11**
3.1. Estimación rango válido de profundidades 12
3.2. Sintonía Poisson para imágenes médicas 13

4. Tracking ORB + Correlación **15**
4.1. Búsqueda activa con puntos ORB 16
4.2. Búsqueda activa por correlación 17

5. Experimentos **20**
5.1. Secuencias 20
5.2. Factores a analizar 22

6. Conclusiones y líneas futuras **25**
6.1. Conclusiones 25
6.2. Líneas futuras 26

7. Bibliografía **27**

Lista de Figuras **29**

Lista de Tablas **30**

Capítulo 1

Introducción y objetivos

1.1. Introducción

Los sistemas de SLAM visual tratan de reconstruir la superficie de una escena y de estimar el movimiento de la cámara respecto de esta. Estos sistemas asumen que la escena que están observando es rígida. Sin embargo, si la escena observada se deforma, la mayoría de los sistemas de SLAM actuales fallarían drásticamente, lo que limita la aplicación de estos algoritmos en escenas donde la deformación es predominante [1] [2].

Uno de los principales campos en los que resulta interesante aplicar este tipo de sistemas es en la medicina, ya que mediante técnicas actuales como la endoscopia es posible la inspección visual de cavidades corporales. Actualmente, durante el proceso de la endoscopia, se capturan imágenes del interior del paciente, se muestran en pantalla y se desechan. El incluir el sistema SLAM permitiría procesarlas para obtener más información, consiguiendo un mapa 3D y la localización del endoscopio respecto de la anatomía del paciente en tiempo real.

Esta información es imprescindible en el caso de que se quiera realizar realidad aumentada sobre los tejidos. También sería necesaria para conseguir la robotización autónoma en el interior del cuerpo, ya que se necesita un mecanismo que informe al robot de dónde está en relación al entorno.

Sin embargo, en el interior del cuerpo los órganos experimentan pequeñas deformaciones debido a los procesos fisiológicos propios de los seres vivos, como la digestión y absorción de alimentos, la contracción muscular o la respiración. Al observar este tipo de escenas con un sistema SLAM monocular rígido, el sistema probablemente terminaría fallando y perdiéndose, por lo que sería más apropiado aplicar un sistema capaz de reconocer las deformaciones como se propone en este trabajo.

Mediante este proyecto se busca estimar la superficie deformable de una escena médica y la pose de la cámara respecto de la escena. Las escenas corresponden con

imágenes reales de endoscopias de animales in-vivo. Para ello, se parte de un software ORBSLAM monocular que ya opera sobre un prototipo de laboratorio donde la escena deformable es una tela que se observa con una cámara. Es la primera vez que se procesan con este software escenas médicas de organismos vivos. El sistema final se prueba sobre un dataset estandarizado: el "Hamlyn dataset" [3].

El dato de entrada al programa es un vídeo de una exploración endoscópica. El endoscopio se mueve y la escena se deforma. El objetivo es estimar, para cada frame del vídeo, la posición del endoscopio y la forma 3D de la superficie. A partir de una exploración rígida inicial, se genera un mapa de puntos en 3D mediante el que se estima una plantilla de la superficie en reposo de la escena. Con esta plantilla se modela una malla triangular sobre la que se estimarán las deformaciones.

El sistema inicial presenta dos barreras principales a la hora de procesar las imágenes médicas. La primera está relacionada con la creación de la malla: en la reconstrucción de la plantilla aparecen puntos espurios que no se encuentran en la escena, dando lugar a mallas erróneas. Los puntos espurios son aquellos que aparecen en la reconstrucción pero no pertenecen a la escena. El segundo obstáculo es la falta de repetitividad que tienen los puntos ORB utilizados en el programa para buscar correspondencias, lo que conlleva un bajo porcentaje de emparejamientos.

1.2. Objetivos

Los objetivos del trabajo son superar las barreras encontradas y evaluar experimentalmente los resultados conseguidos. Para ello, se ha diseñado un mecanismo que establece para los puntos creados un intervalo de profundidades válidas, consiguiendo así que la malla se ajuste correctamente a la escena. También se ha implementado una etapa adicional de búsqueda de emparejamientos por correlación. Así, se logra incrementar el número de correspondencias encontradas en cada imagen y la repetitividad del sistema.

Para analizar las mejoras conseguidas en el procesamiento de imágenes médicas, se procesan tres vídeos de endoscopias in-vivo. Se comprueban experimentalmente los resultados con un ground truth obtenido de las imágenes estéreo.

1.3. Estructura del documento

Para explicar las mejoras aplicadas, a lo largo del documento nos focalizamos en las imágenes de una de las secuencias del "Hamlyn dataset" (figura 1.1). Más adelante,

en el capítulo 5, se detallan los distintos experimentos realizados sobre otros vídeos del dataset.

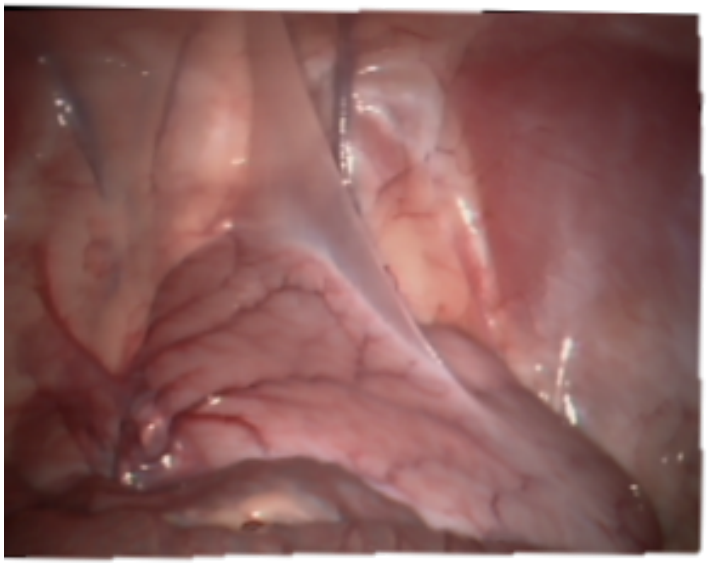


Figura 1.1: Imagen del vídeo al que se hará referencia a lo largo del documento

El documento se estructura de la siguiente forma:

- En el capítulo 2 se explica el funcionamiento del software utilizado, centrándose en los procedimientos utilizados para crear una malla que modele la superficie de la escena observada a partir de una nube de puntos. También se describe como se realiza la optimización SLAM de esta para conseguir que siga las deformaciones que se producen en la escena.
- Más adelante, se detallan los cambios y mejoras que ha habido que realizar en el programa respecto de la versión inicial para conseguir que la malla se adapte correctamente a las escenas médicas (capítulo 3).
- En el capítulo 4 se abordan los problemas surgidos durante el tracking, las mejoras propuestas y los resultados de éstas.
- Finalmente, se exponen los experimentos realizados con tres vídeos seleccionados del dataset y se analizan los resultados con la versión final del software (capítulo 5).

Capítulo 2

SLAM Monocular en escenas deformables

En este capítulo se explica el funcionamiento del sistema ORBSLAM deformable utilizado para procesar los vídeos [4].

Es un método de plantilla que parte de una exploración de la escena en reposo, sin deformación, creando un mapa de puntos en 3D. Esta exploración inicial se lleva a cabo mediante un sistema de ORBSLAM rígido [1].

Tras crear el mapa de puntos, se estima la plantilla (superficie 3D en reposo) a partir de la cual se calculan las deformaciones. En este caso, al ser imágenes médicas, la escena no está en reposo en ningún momento debido a fenómenos como la respiración o la digestión, pero asumimos que sí que lo está hasta que se forma la malla.

2.1. Modelado de la malla

La malla se compone de un conjunto de nodos que se conectan entre sí para definir facetas triangulares. Para crearla, se parte de una nube de puntos dispersa obtenida del SLAM rígido inicial. Estos puntos se encuentran aproximadamente sobre la superficie que queremos obtener. Se usa la técnica de reconstrucción de superficie Poisson como se propone en [5] para construir la malla triangular a partir de la nube de puntos poco densa.

Una vez generada la plantilla, solo los puntos que quedan cercanos a una faceta son conservados. Después, son proyectados en las facetas para forzarlos a que efectivamente se encuentren sobre la superficie. Estos puntos seleccionados son las observaciones que se proyectan en la malla. Finalmente, solo las facetas que contienen algún punto y sus vecinas se mantienen en el modelo final de la superficie.

Al observar la malla con la cámara no se detectan sus nodos, sino que se detectan los puntos del mapa procedentes de la exploración inicial dentro de las facetas. Algunas facetas pueden contener varios puntos observables y otras ninguno. La faceta f_j, que contiene el punto X_j, se define por sus tres nodos $V_{f_j} = \{V_{f,h}\}\ h = 1, 2, 3$.

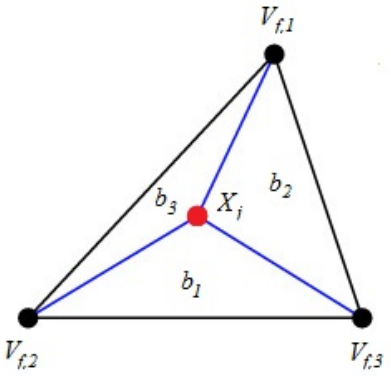


Figura 2.1: Punto observado X_j en la faceta f_j

Para localizar un punto $\mathbf{X}_j$ en la imagen i con respecto a los nodos de su faceta f_j, usamos una interpolación lineal por partes a través de las coordenadas baricéntricas $\mathbf{b}_j = [b_{j,1}, b_{j,2}, b_{j,3}]^\top$ mediante la función $\varphi : [\mathbb{R}^3, \mathbb{R}^{3x3}] \to \mathbb{R}^3$:

$$\mathbf{X}_j^i = \varphi(\mathbf{b}_j, V_{f_j}^i) = \sum_{h=1}^{3} b_{j,h} \mathbf{V}_{f_j,h}^i \tag{2.1}$$

Por lo tanto, cuando la estimación de un nodo cambie durante la optimización, las estimaciones de todos los puntos del mapa que estén asociados a ese nodo cambiarán con él.

El punto observable $\mathbf{X}_j$ definido en $\mathbb{R}^3$ es visto en la imagen i con la cámara localizada en la pose $\mathbf{T}_i$, a través de la función de proyección $\pi : [SE(3), \mathbb{R}^3] \to \mathbb{R}^2$.

$$\pi(\mathbf{T}_i, \mathbf{X}_j^i) = \begin{bmatrix} f_u \frac{x_j^i}{z_j^i} + c_u \\ f_v \frac{y_j^i}{z_j^i} + c_v \end{bmatrix} \tag{2.2}$$

$$[x_j^i\ y_j^i\ z_j^i]^\top = \mathbf{R}^i \mathbf{X}_j^i + \mathbf{t}_i \tag{2.3}$$

Donde $\mathbf{R}^i \in SO(3)$ y $\mathbf{t}^i \in \mathbb{R}^3$ son respectivamente la rotación y traslación de la transformación $\mathbf{T}_i$, y $\{f_u, f_v, c_u, c_v\}$ son las distancias focales y el centro óptico que definen la calibración de la cámara.

2.2. Optimización de la deformación de la malla y posición de la cámara

La optimización se realiza mediante un algoritmo en dos pasos. En el primer paso, se optimizan únicamente los nodos cuyas facetas contienen algún punto observable en la imagen actual y los vecinos de estos (local map $\mathcal{L}_i$). Después, en el segundo paso, la optimización se extiende a todos los puntos del mapa.

El algoritmo utilizado para optimizar tanto la pose de la cámara T_i como la posición de los nodos V_k^i en la imagen i es el siguiente:

$$\begin{aligned}\underset{T_i,V_k^i}{\arg\min}\quad & \frac{1}{N_\bullet}\sum_j \rho\left(\left\|\pi_i\left(T_i,\varphi(\mathbf{b}_j,V_{f_j}^i)\right)-x_j^i\right\|^2\right)\\ &+\frac{\lambda_d}{N_\bullet}\sum_k\sum_{l\in\mathcal{N}_k}\left(\frac{\|V_k^i-V_l^i\|-\|V_k^0-V_l^0\|}{\left\|V_j^0-V_l^0\right\|}\right)^2\\ &+\frac{\lambda_L}{N_\bullet}\sum_k\left(\|\delta_k^i\|-\|\delta_k^0\|\right)^2\sum_{l\in\mathcal{N}_k}\frac{1}{\left\|V_j^0-V_l^0\right\|^2}\\ &+\frac{\lambda_T}{SN_\bullet}\sum_k\left\|V_k^i-V_k^{i-1}\right\|\end{aligned} \tag{2.4}$$

En la ecuación (2.4) encontramos dos grupos de sumandos. El primero es el término de los datos, que es el error de de reproyección. Como el problema es indeterminado, es decir, existen infinitas soluciones que dan residuo 0, se añade un regularizador.

El regularizador penaliza la energía de deformación respecto del reposo. Está inspirado en la mecánica de medios continuos, donde los cuerpos se deforman generando energías internas debido a tensiones normales y tangenciales. Su primer término es la deformación de Cauchy y hace referencia a la deformación longitudinal:

$$\sum_k\sum_{l\in\mathcal{N}_k}\left(\frac{\|V_k^i-V_l^i\|-\|V_k^0-V_l^0\|}{\|V_k^0-V_l^0\|}\right)^2 \tag{2.5}$$

Penaliza la energía de deformación normal. Para cada nodo V_k^i se calcula la deformación respecto a su anillo de nodos vecinos $\mathcal{N}_k$. Es una magnitud adimensional, invariante respecto al tamaño de la faceta.

El segundo regularizador hace referencia a la curvatura (bending):

$$\sum_k\left(\|\delta_k^i\|-\|\delta_k^0\|\right)^2\sum_{l\in\mathcal{N}_k}\frac{1}{\|V_j-V_l\|^2} \tag{2.6}$$

Penaliza energía de deformación tangencial. La formulación de la curvatura es capaz de entender que la malla tiene triángulos irregulares. También es independiente del tamaño de la faceta.

El último término de la ecuación (2.4) es un suavizador temporal entre los nodos del local map $\mathcal{L}_i$. Representa la media de las longitudes de los arcos en la malla. Se adimensionaliza con el factor de escala S, calculado como la distancia mediana entre dos nodos consecutivos.

Los pesos de los regulizadores $\lambda_L, \lambda_d, \lambda_t$ se definen respecto a un peso unitario para el término de datos. Se realiza una corrección dependiendo del número de sumandos, llamado $N_\bullet$, en la suma de los términos de regularización y una corrección de escala para el término temporal.

La optimización se realiza mediante el algoritmo de Levenberg-Marquardt implementado en la librería g2o [6].

Capítulo 3

Estimación de la plantilla para escenas médicas

En este capítulo se detallan los cambios y mejoras que se han realizado respecto del software inicial para conseguir que la malla se adapte correctamente a escenas in vivo de animales.

Partimos de un programa que se utiliza para procesar una tela [4], mientras que en este proyecto se procesan imágenes médicas. Por ello, se proponen una serie de modificaciones en el algoritmo para que funcione correctamente en las mallas típicas obtenidas:

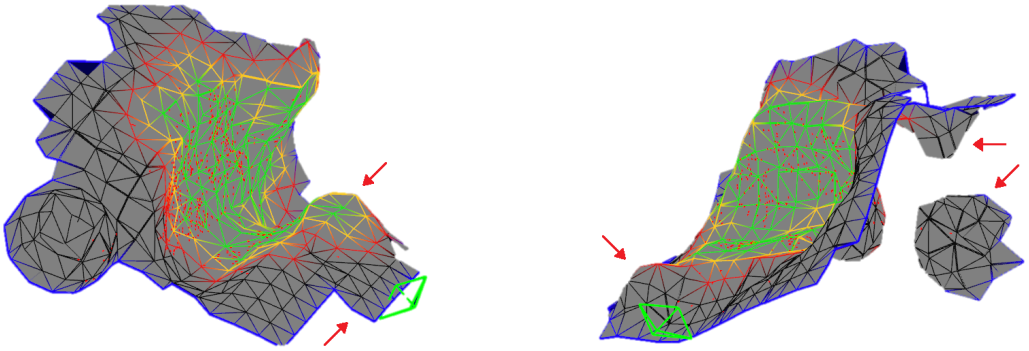
1. Seleccionar cuáles son los puntos que se utilizan para construir la malla, ya que aparecen puntos cercanos y lejanos que producen mallas erróneas.

2. Sintonizar el algoritmo de Poisson para el tipo de mallas obtenidas.

Inicialmente, al correr el programa con las secuencias médicas seleccionadas, se observaba que la malla no tenía la forma esperada y no se adaptaba correctamente a la escena.

Como se puede ver en la figura 3.1, en la malla aparecen dobleces espurios que no corresponden con la escena observada, por ejemplo, un pequeño montículo en la parte más cercana a la cámara. Se observan fragmentos por detrás de la malla principal que no deberían aparecer puesto que las superficies que se están observando en el vídeo son opacas. También aparecen puntos reconstruidos demasiado cerca de la cámara.

Esto se debe a que la plantilla se estima con una nube de puntos que tiene muchos errores. Aparecen puntos mal reconstruidos delante o detrás de la superficie que hacen que la malla no termine de adaptarse a la superficie esperada. Para solucionarlo, se va a filtrar la nube de puntos utilizada para crear la plantilla, eliminando aquellos que estén muy cerca o muy lejos. Como no se tiene escala de la escena, no se puede poner



(a) Vista izquierda (b) Vista derecha

Figura 3.1: Malla errónea. Las flechas rojas indican las partes que no están bien reconstruidas

un umbral fijo que defina lo que está cerca o lejos porque en cada ejecución de un mismo vídeo es distinto. Se va a utilizar un método para medir cuál es la distancia característica a la cámara, y a partir de ella establecer un intervalo de profundidades que se considere válido. Esto se detalla en la sección 3.1. A continuación, se aplica el algoritmo de Poisson como se propone en [4], sintonizándolo para las imágenes médicas. Esta sintonía aparece explicada en la sección 3.2.

3.1. Estimación rango válido de profundidades

Mediante esta mejora se pretende eliminar los puntos espurios, outliers, que aparecen en la reconstrucción e imposibilitan la estimación correcta de la malla. Los outliers se manifiestan como puntos que están demasiado cerca o demasiado lejos de la cámara que observa la superficie. Para ello se implementa un algoritmo que estime, mediante mínima mediana LMedS [7], una distancia característica a la cámara a partir de la que se establece el rango de profundidades aceptado. Este ajuste es necesario porque al utilizar cámaras monoculares, aunque se procese dos veces el mismo vídeo, los umbrales de cercanía y lejanía serán distintos. Tras el ajuste se obtiene la distancia típica de la cámara a la superficie, la cual sirve como medida de escala de la escena.

Para caracterizar la cercanía de un punto, se va a utilizar como parámetro la distancia de este a una de las cámaras que le haya visto. Como la cámara va navegando por la escena, la distancia variará según la cámara escogida para calcularla. Por ello, se establece el criterio de calcular la distancia de un punto como su distancia a la cámara más cercana que le haya visto.

El ajuste por mínima mediana comienza con el cálculo para todos los puntos del

mapa de la distancia a la cámara que le haya visto más cerca. Después, se calcula para cada punto la diferencia entre su distancia a la cámara y la de los demás puntos, y se busca la mediana del residuo cuadrático resultante de esta operación. Finalmente, el punto que ha generado la mínima mediana será utilizado para rechazar los puntos demasiado alejados de él, y pasaremos a considerar su distancia a la cámara como distancia típica de la escena (d_{tip}).

La aceptación o rechazo de un punto viene dado por la desviación típica (s), que se estima a partir de la mínima mediana (eq. 3.1).

$$s = 1{,}4826 \left(1 + \frac{5}{n-k}\right)\sqrt{\text{med}(r_i^2)}\Bigg) \tag{3.1}$$

Donde n es el número total de puntos y k es el número de puntos usados para crear el modelo, en este caso, uno.

Si el residuo de la distancia de cada punto, dividido por la desviación típica, supera un cierto umbral, entonces se rechaza (eq. 3.2). Lo mismo ocurre con los puntos demasiado cercanos a la cámara que quedan por debajo de su umbral (eq. 3.3).

$$\left|\frac{d_i - d_{tip}}{s}\right| > 3 \qquad (3.2) \qquad\qquad \left|\frac{d_i - d_{tip}}{s}\right| < 2 \qquad (3.3)$$

Como se observa en la figura 3.2, no se tienen en cuenta los puntos más cercanos y alejados de la cámara para crear la malla, consiguiendo así una forma mucho más coherente con la superficie esperada.Una vez descartados estos puntos, se pasa a calcular la malla con los puntos restantes del mapa aplicando el algoritmo de Poisson.

3.2. Sintonía Poisson para imágenes médicas

La malla se obtiene utilizando la técnica de reconstrucción de superficie Poisson a partir de la nube de puntos [5]. Para poder usar esta técnica, se necesita una nube de puntos densa con el fin de conseguir un conjunto de normales que varíe suavemente a lo largo de la superficie. Por ello, densificamos nuestra nube de puntos poco densa con el método MLS (Moving Least Squares) [8]. Este algoritmo sobremuestrea el plano local de cada punto de la nube inicial, aproximando la superficie a un polinomio de orden dado, en este caso de orden tres. Se logra una densidad de puntos constante en un radio característico dado alrededor de cada punto del mapa.

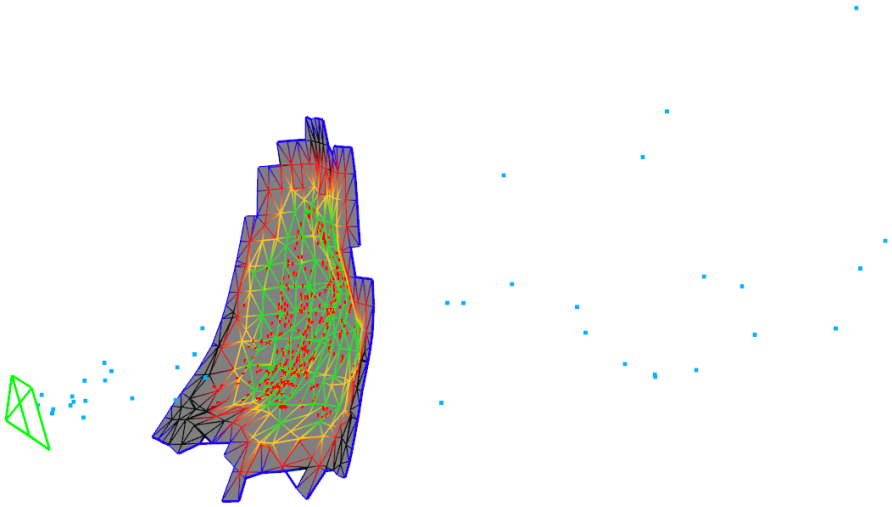


Figura 3.2: Malla tras limitar la profundidad. En azul: puntos eliminados. En rojo: puntos del mapa conservados

Para las imágenes médicas, el algoritmo se ha utilizado con la misma sintonía que en el software inicial, modificando únicamente el valor del radio característico. Este radio tiene que estar relacionado con el tamaño de la escena observada. Por este motivo, se asigna al radio un valor del 50 % de la distancia típica a la cámara (d_{tip}) calculada anteriormente, ya que al estar utilizando imágenes procedentes de cámaras monoculares no se puede conocer el tamaño real de la escena. Se ha escogido esta magnitud porque es la que mejor funciona según se ha comprobado experimentalmente.

Capítulo 4

Tracking ORB + Correlación

En este capítulo se abordan las limitaciones del algoritmo actual de tracking, presentando las técnicas utilizadas para superarlas, así como los resultados conseguidos.

El tracking se encarga de localizar la cámara respecto de cada frame procesado. Para cada frame, se parte de una hipótesis inicial de la posición de la cámara respecto de la malla. A continuación, se predice la posición de los puntos del mapa en la imagen y se proyectan sobre esta. Se calcula una región de búsqueda de 15x15 píxeles alrededor de cada predicción y se busca activamente en ella correspondencias (matches). Finalmente se optimiza la posición de la cámara y la deformación de la malla con los matches encontrados.

El sistema de SLAM empleado en este programa utiliza la búsqueda activa con puntos ORB. La búsqueda por ORB consiste en detectar puntos de interés en la imagen actual, a los que se les asigna un descriptor. Cada punto del mapa proyectado se empareja con el punto de interés que tenga un descriptor más parecido, dentro de su región de búsqueda. Este método se emplea debido a sus múltiples ventajas, por ejemplo, su rapidez en el emparejamiento gracias a que solo procesa los puntos extraídos. Los puntos ORB también juegan un papel muy importante en el caso de que la cámara se pierda, ya que facilitan su relocalización.

Sin embargo, la principal debilidad del algoritmo basado únicamente en ORB es que no es repetible. Esto es debido a que el detector de puntos de interés solo es capaz de emparejar puntos donde haya disparado. Si no dispara en la región en la que se está buscando un punto del mapa, no lo va a encontrar. Interesa una elevada repetitividad para poder realizar predicciones más exactas de las deformaciones que aparecen. Por ello, se propone implementar un paso adicional para conseguir mayor repetitividad: la búsqueda activa por correlación. La búsqueda por correlación funciona de manera distinta a los puntos ORB: tras predecir dónde se encuentra el punto del mapa en la imagen, busca en todos los píxeles de su región de búsqueda alrededor de la predicción

a ver si está ahí. De esta forma, el problema que surgiría con cualquier método basado en la detección de puntos de interés, de que lo que se está intentando emparejar no ha quedado entre los puntos extraídos, desaparece. Es un método más repetible, pero también más caro.

Por lo tanto, la propuesta final consiste en comenzar siempre la búsqueda de correspondencias a partir de los puntos ORB en las regiones de búsqueda de los puntos del mapa proyectados, debido a las ventajas comentadas anteriormente. Después, aquellos puntos del mapa que no hayan podido ser emparejados, se buscarán por correlación para conseguir aumentar la repetitividad de los emparejamientos.

4.1. Búsqueda activa con puntos ORB

Es el método clásico utilizado en el sistema de SLAM. Su funcionamiento básico consiste en que cuando llega una imagen nueva, el detector de puntos extrae aquellos puntos que considera de interés y los empareja con los puntos del mapa.

Este método utiliza el detector de puntos FAST [9], extrayendo puntos de interés en 8 niveles de la pirámide de escala, con un factor de escala de 1.2. En nuestro caso únicamente utilizamos el primer nivel de escala, ya que al considerar más el detector extraía varias veces el mismo punto en diferentes escalas, produciéndose grandes errores de reproyección.

Para buscar puntos de interés, el detector FAST selecciona un punto p en la imagen y verifica si n puntos consecutivos de los que le rodean son más claros o más oscuros que él, con un umbral de diferencia. Si esto ocurre, entonces se considera que el punto p es un punto de interés. Con el objetivo de conseguir una distribución homogénea de puntos, se divide la imagen en una cuadrícula, intentando extraer al menos 5 puntos por celda.

Una vez extraídos los puntos de interés, se calcula la orientación de estos y su descriptor ORB [10]. El descriptor ORB es empleado más adelante en el emparejamiento con los puntos del mapa.

A continuación, se busca cada punto del mapa en la imagen actual. Para ello, se proyecta el punto en la imagen y se calcula a su alrededor la región de búsqueda. Se compara su descriptor con el de los puntos ORB extraídos anteriormente que se encuentran en la región de búsqueda. Si se encuentra uno con un descriptor suficientemente parecido, se empareja con este.

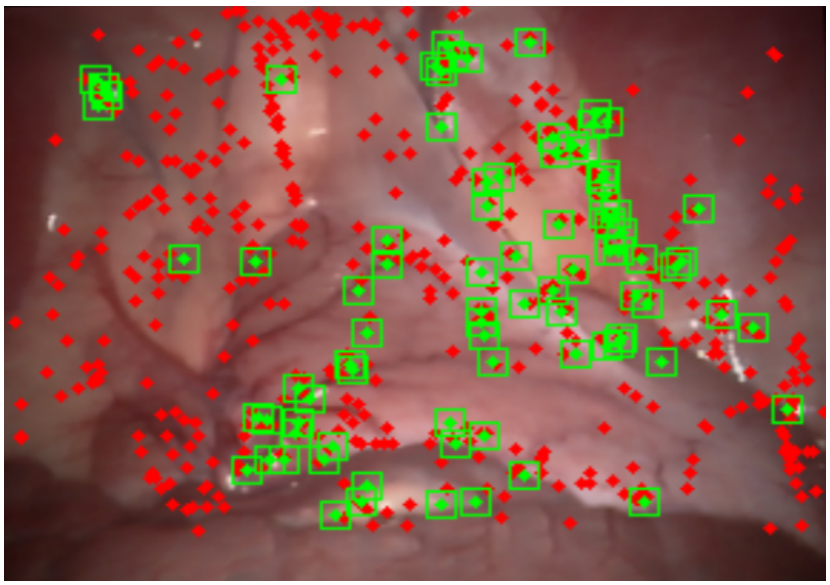


Figura 4.1: Emparejamientos con puntos ORB. En verde: puntos ORB. En rojo: puntos del mapa proyectados.

En la figura 4.1 se observa el problema de repetitividad comentado anteriormente. El detector de puntos no dispara en ciertas zonas, por lo que muchos puntos del mapa se quedan sin emparejar.

4.2. Búsqueda activa por correlación

El funcionamiento de la búsqueda por correlación se basa en reconocer y localizar una subimagen dentro de una imagen. En este caso, se extrae para cada punto del mapa que aparece en la imagen vista y que no ha sido emparejado anteriormente por ORB, un patch de 5x5 píxeles de su keyframe de referencia. Este patch es el que se buscará en la región de búsqueda calculada para cada punto del mapa.

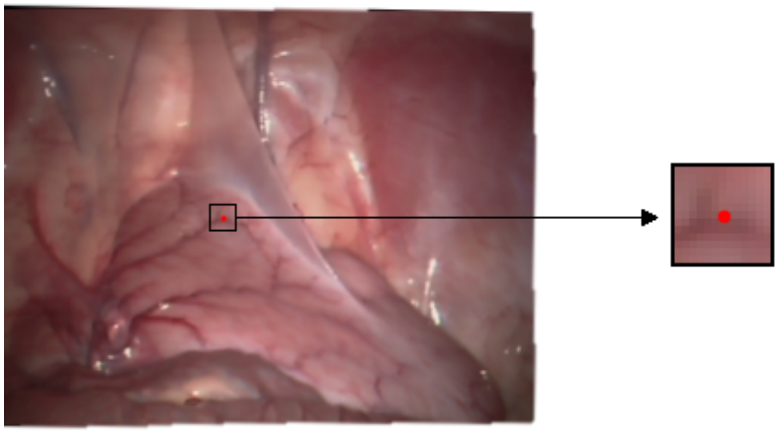


Figura 4.2: Patch asociado a un punto del mapa

Para localizar la posición óptima del patch, se va moviendo este por todos los píxeles de la ventana de búsqueda (de izquierda a derecha y de arriba abajo). En cada posición, se calcula el coeficiente de correlación normalizado ρ (eq. 4.1), que representa cómo de bueno o malo es el match en esa posición o, lo que es lo mismo, cómo de similar es el patch a esa área de la imagen si ese píxel fuese el match.

$$\rho = \frac{\sum_{ij}(W_{ij}^a - \overline{W}_{ij}^a)(W_{ij}^b - \overline{W}_{ij}^b)}{\sqrt{\sum_{ij}(W_{ij}^a - \overline{W}_{ij}^a)^2}\sqrt{\sum_{ij}(W_{ij}^b - \overline{W}_{ij}^b)^2}}[11] \tag{4.1}$$

Donde W_{ij}^a representa la ventana de búsqueda de la imagen principal y W_{ij}^b el patch del punto.

Este coeficiente puede tomar valores entre [-1, 1], siendo la mejor posición aquella en la que se obtenga el máximo valor. Además, también se ha aplicado un umbral para asegurar un buen matching, de forma que solo serán válidos aquellos emparejamientos que obtengan un coeficiente de correlación mayor de 0.8. Si se encuentra un píxel que cumpla estos requisitos, se empareja el píxel con el punto del mapa y se añade a la ecuación de optimización.

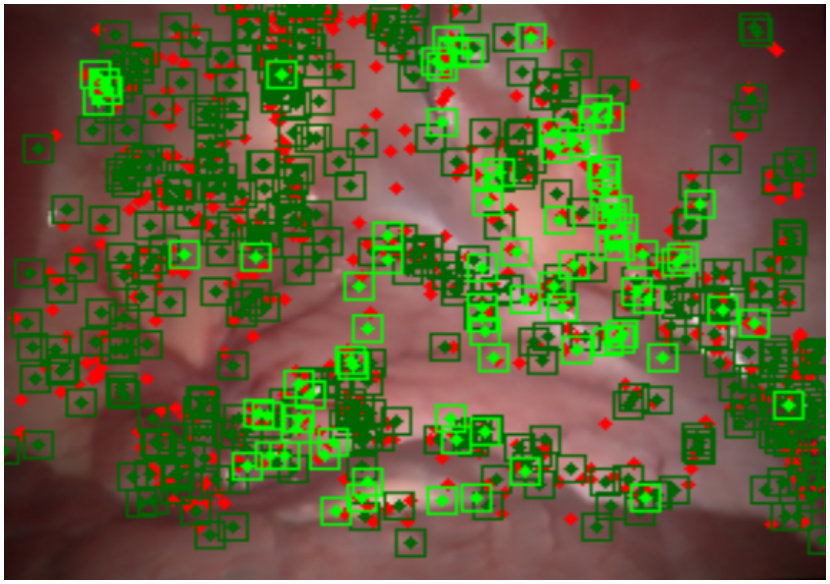
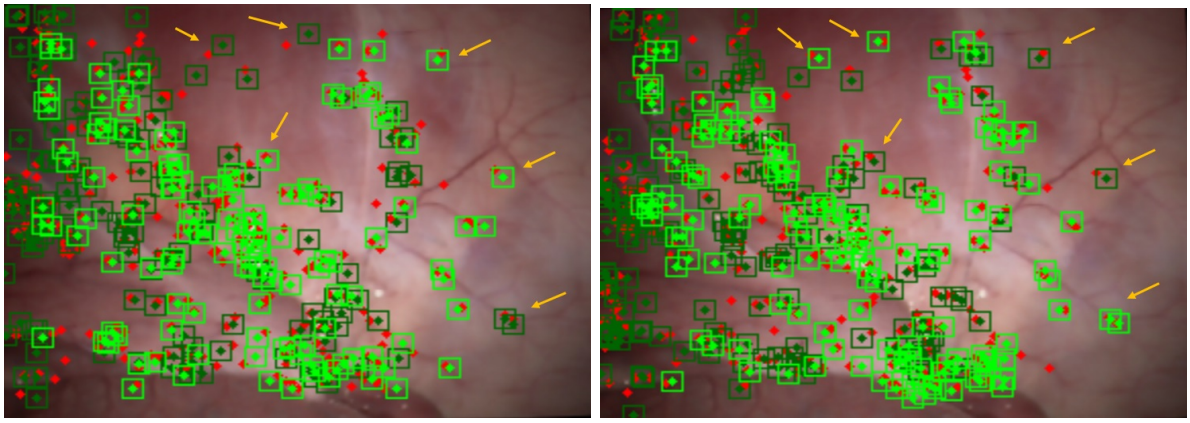


Figura 4.3: Emparejamientos con puntos ORB y por correlación. En verde claro: puntos ORB. En verde oscuro: puntos predichos por correlación. En rojo: puntos del mapa proyectados.

En la figura 4.3 podemos ver como aparecen emparejados por correlación los puntos del mapa que no conseguía emparejar mediante puntos ORB en la figura 4.1.

Si nos fijamos en un punto concreto a lo largo de la secuencia, podemos ver como a veces aparece emparejado con un punto ORB y otras mediante correlación (figura 4.4).

Logramos así un alto porcentaje de emparejamientos en cada frame, y por lo tanto también aumentar la repetitividad.



(a) (b)

Figura 4.4: Mismos puntos del mapa emparejados en imágenes consecutivas por puntos ORB y por correlación. Algunos ejemplos aparecen señalados por una flecha.

Capítulo 5

Experimentos

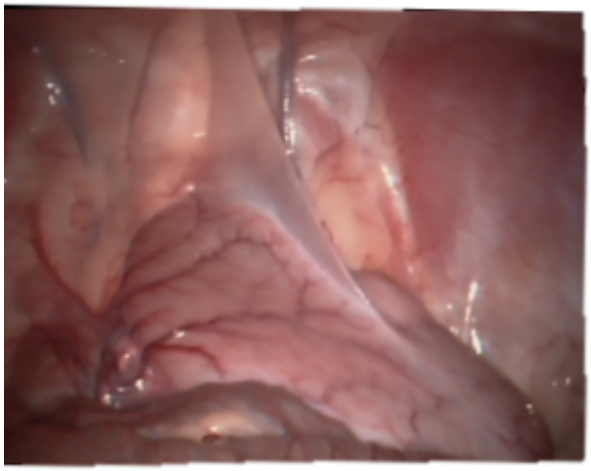
Se han seleccionado del dataset las 3 secuencias consideradas más apropiadas para poder analizar los resultados, teniendo en cuenta los objetivos del proyecto. Las secuencias son endoscopias realizadas in-vivo en cerdos. De todas se dispone del par estéreo para poder estimar el ground truth con el que se evaluará el error cometido en la estimación. Sin embargo, en la secuencia 1 la cámara del endoscopio es distinta a las demás y no hemos conseguido hacer funcionar el algoritmo estéreo.

Para la validación experimental, se van a comparar los resultados obtenidos cuando la búsqueda de emparejamientos se realiza únicamente por puntos ORB y cuando se realiza por ORB y correlación. Las secuencias se van a procesar en ambos casos con el algoritmo ya modificado para obtener una plantilla adecuada, como se ha explicado en el capítulo 3. De no haber aplicado este método para estimar la plantilla las secuencias no habrían podido ser procesadas. Así, el experimento se focaliza en ver el efecto que tiene el cambiar el algoritmo de emparejamiento en el desempeño del sistema.

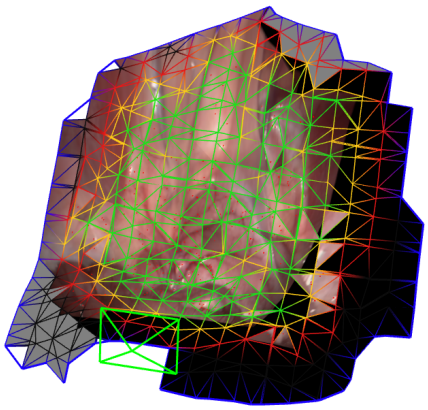
A continuación se exponen las secuencias ordenadas por grado de dificultad. A la izquierda aparece una imagen característica del vídeo, que permite entender cómo es la cavidad. A la derecha se muestra la malla que modela la escena. Los puntos rojos que aparecen en la malla son las proyecciones de los puntos del mapa sobre la imagen.

5.1. Secuencias

Secuencia 1 Esta secuencia presenta una escena con poca deformación, únicamente se observa el pulso. Además, se realiza una exploración inicial por la cavidad con la escena prácticamente en reposo. La cámara realiza un movimiento uniforme a lo largo de la escena: se desplaza de izquierda a derecha de la cavidad y luego vuelve al comienzo. Debido a esto, se consideró una secuencia sencilla con la que comenzar a trabajar para conseguir resultados.



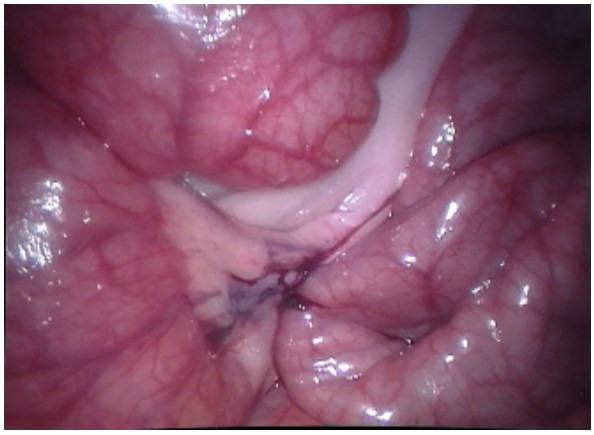
(a) Imagen del vídeo



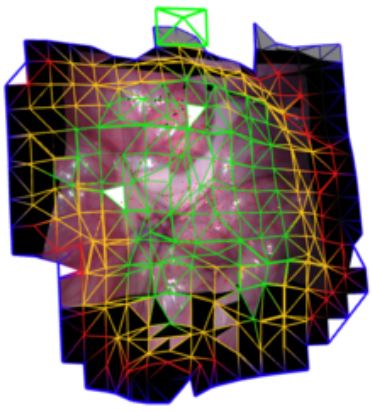
(b) Malla estimada

Figura 5.1: Secuencia 1

Secuencia 2 En esta secuencia la cámara se mueve ligeramente alrededor de la escena de la figura 5.2. Las vísceras experimentan pequeñas deformaciones y aparece también movimiento relativo entre ellas. Existe una exploración inicial pero con deformación, lo que dificulta la creación de la malla.



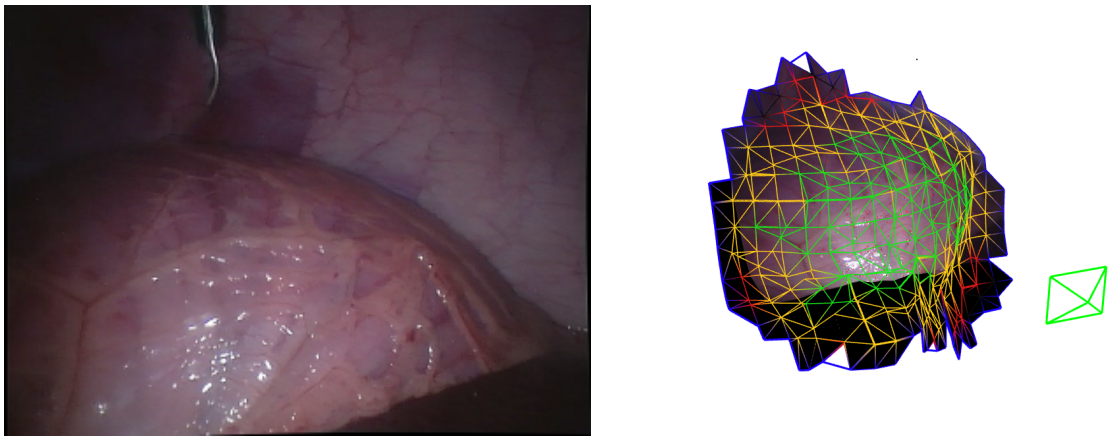
(a) Imagen del vídeo



(b) Malla estimada

Figura 5.2: Secuencia 2

Secuencia 3 Esta secuencia comienza con una exploración inicial de la escena en reposo. A continuación, la superficie es deformada por una pinza. La pinza mueve también una membrana en la superficie, que puede ser problemática. La cámara apenas se mueve por la escena.



(a) Imagen del vídeo (b) Malla estimada

Figura 5.3: Secuencia 3

5.2. Factores a analizar

Para valorar el efecto que tiene añadir en el tracking la búsqueda por correlación, se van a analizar los siguientes factores:

Número de puntos emparejados: se calculan en porcentaje respecto de los puntos predichos en cada frame. Este dato cuantifica la repetitividad conseguida. También se han tomado medidas del número de matches totales.

Tiempo de cómputo: midiendo los tiempos que cuesta realizar los emparejamientos podemos comparar cómo de caro resulta el método que utiliza tanto búsqueda por ORB como por correlación, frente al que únicamente busca puntos ORB.

En la tabla 5.1, aparece para cada uno de los métodos (ORB y correlación) el tiempo total que le lleva realizar todos los emparejamientos en una imagen. Conociendo el número de puntos del mapa predichos en la imagen y que por lo tanto son candidatos a ser emparejados, se calcula el tiempo por candidato (tiempo relativo respecto al número de puntos predichos). Además, en la columna de ORB+Correlación, se incluye el tiempo total de la etapa de emparejamiento empleando los 2 métodos.

Error estimación superficie de la escena se calcula con el algoritmo estéreo un ground truth con el que se estima el error cometido. En la escena de la secuencia 1 no funciona porque está tomada con un endoscopio distinto. Se calcula tanto el error mediano como el error máximo para ambos casos.

La tabla 5.1 recoge los valores de estos factores para las diferentes secuencias. Debido

	ORB					ORB+Correlación					
	Nº matches/ % matches	Tiempo (ms)		Error (mm)		Nº matches/ % matches	Tiempo (ms)			Error (mm)	
		rel	tot	med	máx		corr rel	corr tot	tot	med	máx
Secuencia 1	140/ 36 %	0.0057	2.19	-	-	321/ 95 %	0.036	7.25	9.39	-	-
Secuencia 2	171/ 32 %	0.0048	2.32	3.42	29.80	471/ 97 %	0.034	11.53	13.87	5.08	28.75
Secuencia 3	56/ 31 %	0.0054	1.05	4.13	36.75	168/ 96 %	0.037	4.43	5.45	5.55	38.90

Tabla 5.1: Comparación entre los métodos de emparejamiento

a la variación de los resultados entre distintas ejecuciones, se ha procesado 5 veces cada vídeo y se ha hecho una media con los datos obtenidos.

A partir de los resultados obtenidos en la tabla podemos sacar diversas conclusiones. Si nos fijamos en el número de matches encontrados, cuando estos se emparejan únicamente por ORB se consiguen porcentajes muy bajos, en torno a un tercio de los puntos predichos. Este valor demuestra la baja repetitividad que tienen los puntos ORB.

Sin embargo, al añadir la búsqueda por correlación, el número de correspondencias encontradas se incrementa significativamente, consiguiendo valores próximos al 100 %, lo que supone emparejar la mayoría de los puntos posibles de la imagen.

En cuanto a los tiempos, se observa que el método de búsqueda por correlación es claramente más caro, como ya se había predicho. El tiempo por candidato es unas 6 veces mayor que en la búsqueda por ORB. No obstante, habría que tener en cuenta también el tiempo que cuesta extraer los puntos ORB, que es aproximadamente unos 10 ms por imagen. Si consideramos este tiempo, vemos que el tiempo de los emparejamientos por ORB es despreciable y que la de los emparejamientos por correlación es similar al tiempo de detección. Por lo tanto, el costo total de los emparejamientos por ORB y que la de los emparejamientos por correlación es similar.

Estos tiempos corresponden a la etapa del procesamiento de imagen. Después comienza la optimización no lineal, que es muy costosa. El tiempo total de tracking de un frame suele encontrarse alrededor de los 700 ms, mucho mayor que el total de la extracción y la búsqueda de emparejamientos. Por lo tanto, el tiempo invertido en la búsqueda por correlación es asumible.

En relación a los errores calculados, los resultados obtenidos en ambos casos son muy buenos, del orden de pocos milímetros. Si los comparamos, vemos que después de la correlación el error mediano es ligeramente mayor, pero son comparables. No se ha mejorado, pero se consigue mantener al incrementar el número de matches casi al triple. Como se recuperan muchos más puntos que antes, se está dando mayor información

al sistema por lo que los resultados son más precisos. Además, al estar midiendo más puntos, el regularizador de la ecuación (2.4) tiene que afectar a zonas más grandes de la escena.

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

Se han procesado por primera vez con el software de ORB-SLAM monocular deformable vídeos médicos, procedentes de un dataset público, de endoscopias realizadas a animales in-vivo.

El software no se adaptaba bien a las imágenes médicas, por lo que no se conseguían resultados coherentes. Únicamente estableciendo un intervalo de profundidades válidas estimado con LMedS se consiguió que la malla se adaptase al tipo de escenas procesadas (comprobación visual). Al añadir los matches por correlación, se logra incrementar notablemente el número de emparejamientos.

Se ha añadido una etapa que permite calcular un ground truth para comparar los resultados. El sistema es capaz de mantener el mismo error de estimación de la superficie que cuando solo se utilizaban los puntos ORB, pero en este caso consiguiéndose emparejar muchos más puntos por lo que se le está dando más información al sistema.

Se ha diseñado todo el sistema necesario para poder hacer una sintonía final de los pesos de los regularizadores, y así conseguir una reconstrucción mucho más precisa, que es sin duda el objetivo último del sistema.

Aunque se busca conseguir la automatización completa de los procesos, el sistema también funcionaría mejor si se realizase un tuning de los distintos regularizadores para cada secuencia procesada. Gracias al LMedS, se ha simplificado la sintonía fina de los rangos de profundidades que permiten optimizar la adaptación de la malla a la forma de la escena.

6.2. Líneas futuras

A corto plazo, habría que realizar mucha más experimentación con el software para conseguir mayor robustez. También sería interesante tratar secuencias de diferentes procedimientos quirúrgicos para identificar los auténticos retos que supone la cirugía sobre humanos.

El campo del SLAM deformable es un terreno en el que todavía queda mucho por investigar. En cualquier caso hoy por hoy es la única técnica para obtener, a frecuencia de vídeo, una estimación de la posición del endoscopio respecto de la escena. Esta estimación es obligada para incluir realidad aumentada y robotización en los procedimientos quirúrgicos cotidianos. Una primera línea sería aplicarlo para añadir anotaciones de realidad aumentada tales como información preoperatoria. En un futuro más lejano permitiría la robotización de procesos quirúrgicos dentro del cuerpo donde la información SLAM es imprescindible.

Capítulo 7

Bibliografía

[1] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[2] Nader Mahmoud, Toby Collins, Alexander Hostettler, Luc Soler, Christophe Doignon, and Jose Maria Martinez Montiel. Live tracking and dense reconstruction for hand-held monocular endoscopy. *IEEE Transactions on Robotics*, 2018.

[3] Peter Mountney, Danail Stoyanov, and Guang-Zhong Yang. Three-dimensional tissue deformation recovery and tracking. 2010. Available at http://hamlyn.doc.ic.ac.uk/vision/.

[4] Jose Lamarca and Jose Maria Martinez Montiel. Camera tracking for SLAM in deformable maps. In *4th International Workshop on Recovering 6D Object Pose, ECCV Workshop*, 2018.

[5] M. Kazdhan and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70, 2006.

[6] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE, 2011.

[7] Peter J. Rousseeuw. Least median of squares regression. *American Statistical Association*, 79(388):871–880, 1984.

[8] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Schachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, 2003.

[9] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[10] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFTor SURF. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[11] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

Lista de Figuras

1.1. Imagen del vídeo al que se hará referencia a lo largo del documento . . 6

2.1. Punto observado X_j en la faceta f_j 8

3.1. Malla errónea. Las flechas rojas indican las partes que no están bien reconstruidas 12
3.2. Malla tras limitar la profundidad. En azul: puntos eliminados. En rojo: puntos del mapa conservados 14

4.1. Emparejamientos con puntos ORB. En verde: puntos ORB. En rojo: puntos del mapa proyectados. 17
4.2. Patch asociado a un punto del mapa 17
4.3. Emparejamientos con puntos ORB y por correlación. En verde claro: puntos ORB. En verde oscuro: puntos predichos por correlación. En rojo: puntos del mapa proyectados. 18
4.4. Mismos puntos del mapa emparejados en imágenes consecutivas por puntos ORB y por correlación. Algunos ejemplos aparecen señalados por una flecha. 19

5.1. Secuencia 1 21
5.2. Secuencia 2 21
5.3. Secuencia 3 22

Lista de Tablas

5.1. Comparación entre los métodos de emparejamiento 23