Universidad Zaragoza
1542

Trabajo de Fin de Grado en Ingeniería de Tecnologías Industriales

SLAM visual estéreo aplicado a endoscopias médicas

Autor

Ignacio Cuiral Zueco

Director

José María Martínez Montiel

Escuela de Ingeniería y Arquitectura

2017

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. Ignacio Cuiral Zueco,

con nº de DNI 73028532 J 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) SLAM visual estéreo aplicado a endoscopias médicas,

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

Zaragoza, 21 de septiembre de 2017

Fdo: Ignacio Cuiral Zueco

SLAM visual estéreo aplicado a endoscopias médicas

RESUMEN

ORBSLAM es un software que, a partir de una secuencia en vídeo grabada por una cámara que se mueve por una escena genera un mapa de puntos tridimensional (no denso) y permite la localización de la cámara con respecto de la escena a tiempo real. ORBSLAM presenta la posibilidad de una entrada de par estéreo, es decir, dos cámaras. En el presente trabajo se realiza una evaluación de la viabilidad del módulo Stereo de ORB-SLAM2 aplicado a escenas de endoscopias y laparoscopias médicas. Además, aprovechando uno de los puntos fuertes del estéreo, como más adelante se detallará, se ha ampliado el módulo Stereo del programa ORBSLAM añadiendo la posibilidad de generar un mapa denso tridimensional de la escena en observación.

Se ha hecho una valoración experimental de ORBSLAM sobre pares estéreo en endoscopias porcinas y una densificación de la anatomía porcina observable en las secuencias, alcanzando prestaciones a tiempo real.

Índice

1. Introducción y objetivos **6**

1.1. Introducción 6

1.2. Objetivos 7

1.3. Estructura del trabajo 8

2. SLAM con mapa disperso, comparación estéreo frente monocular **9**

2.1. ORBSLAM monocular 9

2.2. ORBSLAM stereo 12

2.3. Comparación entre SLAM monocular y stereo 14

2.4. Re-localización y cerrado de bucles 15

2.4.1. Concepto de KeyFrame 15

2.4.2. Re-localización 16

2.4.3. Cerrado de bucles 16

3. Densificación en ORBSLAM **18**

3.1. Densificación de un par estéreo: mapa denso local 18

3.1.1. Geometría epipolar y concepto de disparidad 18

3.1.2. Rectificación de las secuencias de la base de datos de Hamlyn . 19

3.1.3. Generación del mapa de disparidades para un par estéreo 20

3.1.4. Generación de los puntos 3D obtenidos a partir del mapa de disparidades. 22

3.2. Mapa denso global 23

3.2.1. Alineación de los mapas densos locales 23

3.2.2. Rechazo de los puntos densos duplicados 23

3.3. Conclusiones 23

4. Resultados experimentales **24**

4.1. Organización y acondicionamiento de los datos del Hamlyn Centre . . . 24

4.2. Sintonía de ORBSLAM 26

4.3. Análisis de las secuencias y evaluación de ORBSLAM 27

4.3.1. Secuencia 1 . . . 29
4.3.2. Secuencia 2 . . . 30
4.3.3. Secuencia 3 . . . 31
4.3.4. Secuencia 4 . . . 32
4.3.5. Secuencia 5 . . . 33
4.3.6. Secuencia 6 . . . 34
4.4. Conclusiones . . . 36
4.5. Tiempos de cómputo . . . 36
4.5.1. Tablas . . . 36
4.5.2. Análisis y conclusiones de los tiempos de cómputo . . . 37

5. Líneas futuras **39**
5.1. Corto plazo . . . 39
5.2. Largo plazo . . . 40

6. Bibliografía **42**

Anexos **44**

A. Ejemplo de un "hamlyn.yaml". **45**

B. Repositorio Github **47**

Lista de Figuras **48**

Lista de Tablas **50**

Capítulo 1

Introducción y objetivos

1.1. Introducción

Los sistemas de VSLAM (Simultaneous Localization and Maping, from Visual sensors) emplean como dato de entrada únicamente la secuencia de imágenes tomada por una cámara móvil que observa una escena, el objetivo es estimar simultáneamente un mapa 3D de la escena observada y la trayectoria seguida por la cámara. Su investigación comenzó en las últimas décadas del siglo XX dentro de campo de la robótica móvil. La evolución de las cámaras digitales y la normalización de su uso en dispositivos móviles han potenciado el interés comercial por los sistemas VSLAM. Se dispone de cámaras más pequeñas, más baratas, y de mayor resolución, que abren las puertas a nuevas áreas de aplicación. Actualmente es una técnica madura que está empezando su migración a las aplicaciones comerciales, no sólo en robótica sino también en realidad virtual y aumentada.

El uso de las cámaras es prevalente en la exploración médica. En muchos casos sólo se dispone de la información que proporcionan las cámaras ya que otros sensores son inviables. La técnica diagnóstica de la endoscopia permite la inspección visual de cavidades corporales así como intervenciones poco invasivas. Actualmente, durante el proceso de la endoscopia, se capturan las imágenes del interior del paciente, se muestran en pantalla y se desechan tras su observación. El incluir el sistema SLAM implicaría capturar las imágenes y además procesarlas para obtener más información. Se conseguiría así un mapa 3D y localización respecto de la anatomía del paciente en tiempo real. Además, al utilizar hoy en día equipos ya informatizados, únicamente sería necesario añadir un módulo de software.

Se dispone de un sistema SLAM estéreo, ORB-SLAM2 [1], estado del arte para aplicaciones de robótica. Este sistema ya ha sido evaluado en endoscopias [2] en su versión monocular [3]. En este trabajo se van a evaluar sus prestaciones en el procesamiento de secuencias de endoscopias médicas en estéreo.

El SLAM monocular presenta potencial para la miniaturización del endoscopio, pero tiene robustez limitada con respecto a la inicialización y frente a la deformación de la escena. Además, el proceso de obtención de un mapa denso de la escena en monocular es complejo. El estéreo presenta inconvenientes para la miniaturización porque las dos cámaras deben tener separación suficiente entre ellas pero, por otro lado, el conocer la magnitud de esa separación permite recuperar la escala real de la escena evitando también su deriva [4]. En estéreo la información densa de la profundidad de la escena está disponible de forma inmediata para cada fotograma, además tiene una gran robustez en la inicialización y frente a la deformación de la escena.

ORBSLAM genera un mapa "sparse" (no denso) que localiza la cámara con precisión pero produce un mapa de puntos de la escena pobre. El trabajar en estéreo permite realizar observaciones densas de zonas parciales de la escena.Se propone alinear todas las observaciones densas obtenidas en estéreo a partir de la posición de la cámara que proporciona el mapa sparse.

1.2. Objetivos

En este trabajo se va a hacer una valoración de ORB-SLAM2 Stereo aplicado a endoscopias así como una reconstrucción 3D densa del interior del paciente como aplicación práctica de una de las ventajas que ofrece el sistema estéreo.

Se analizarán secuencias estéreo de endoscopias obtenidas de la base de datos del Hamlyn Center de Londres. Este dataset es público y standard, de forma que la comparación con otros sistemas SLAM es repetible.

Los objetivos son:

- Adaptación del Hamlyn dataset para su procesamiento con el sistema ORBSLAM. Esta adaptación incluye rectificación de las imágenes y acondicionamiento de los parámteros de calibración.

- Procesamiento del dataset con ORBSLAM Stereo (mapa sparse). Sintonía del programa y evaluación de la calidad de los resultados y su costo computacional.

- Densificación del mapa. Extracción de información densa en bruto de los pares estéreo, alineamiento de las observaciones densas parciales en el mapa global y evaluación de la calidad y del costo computacional.

- Creación de un repositorio en "Github" con las modificaciones realizadas en el código de ORBSLAM (anexo B).

1.3. Estructura del trabajo

La estructura del trabajo es la siguiente:

- Presentación de los sistemas SLAM estéreo y monocular y posterior comparación entre ellos.
- Densificación de un fotograma estéreo, almacenamiento y alineación de las diferentes reconstrucciones densas para la formación de un mapa denso global.
- Proceso de adaptación del Hamlyn dataset para el procesamiento de sus secuencias.
- Resultados experimentales de las 6 secuencias más representativas.
- Discusión y trabajo futuro.

Se incluyen anexos como complemento, con información considerada pertinente.

Capítulo 2

SLAM con mapa disperso, comparación estéreo frente monocular

En este capítulo se explican brevemente los sistemas ORBSLAM monocular y ORBSLAM stereo, para posteriormente, realizar una comparación entre ambos.

2.1. ORBSLAM monocular

Se hace frente al reto de la generación de un mapa de la escena en observación y de localización a tiempo real con respecto a dicho mapa y, por lo tanto, con respecto de la escena con el uso de una sola cámara, es decir, monocular.

Para resolver dicho problema ORBSLAM utiliza un proceso denominado Bundle Adjustment (BA) o Ajuste de Haces que se basa en la búsqueda de correspondencias de puntos de interés (observaciones) en un set o grupo de imágenes de una misma escena. Se entiende por punto o región de interés a aquellas zonas de una imagen que contienen información potencialmente rastreable como es, por ejemplo, una zona con un cambio brusco de iluminación, una esquina de un objeto, etc. Los haces son las rectas definidas por el punto en el espacio 3D y el centro óptico de cada cámara que lo observa.

Al trabajar en monocular el set o grupo de imágenes se obtendría desplazando una cámara por diferentes regiones de la escena. Se trata, a continuación, de localizar los puntos de interés en diferentes imágenes buscando correspondencias de un mismo punto 3D visto desde diferentes puntos de vista (figura 2.1). Esto implica ver un punto desde diferentes posiciones y orientaciones y, por lo tanto, obtener diferentes haces para un mismo punto tridimensional de la escena. Cuando se dan correspondencias de puntos vistos en varias imágenes diferentes se resuelve la geometría de esa parte de la escena y se añaden los puntos al Map Point (o mapa de puntos 3D). También se añaden

la posición y la orientación que la cámara tenía en cada uno de dichos fotogramas a la trayectoria general de la cámara. Así se va generando el mapa de puntos y se va obteniendo la trayectoria de la cámara simultáneamente. El método que utiliza ORBSLAM para la búsqueda de puntos de interés y de sus correspondencias en las imágenes es el método ORB [5].

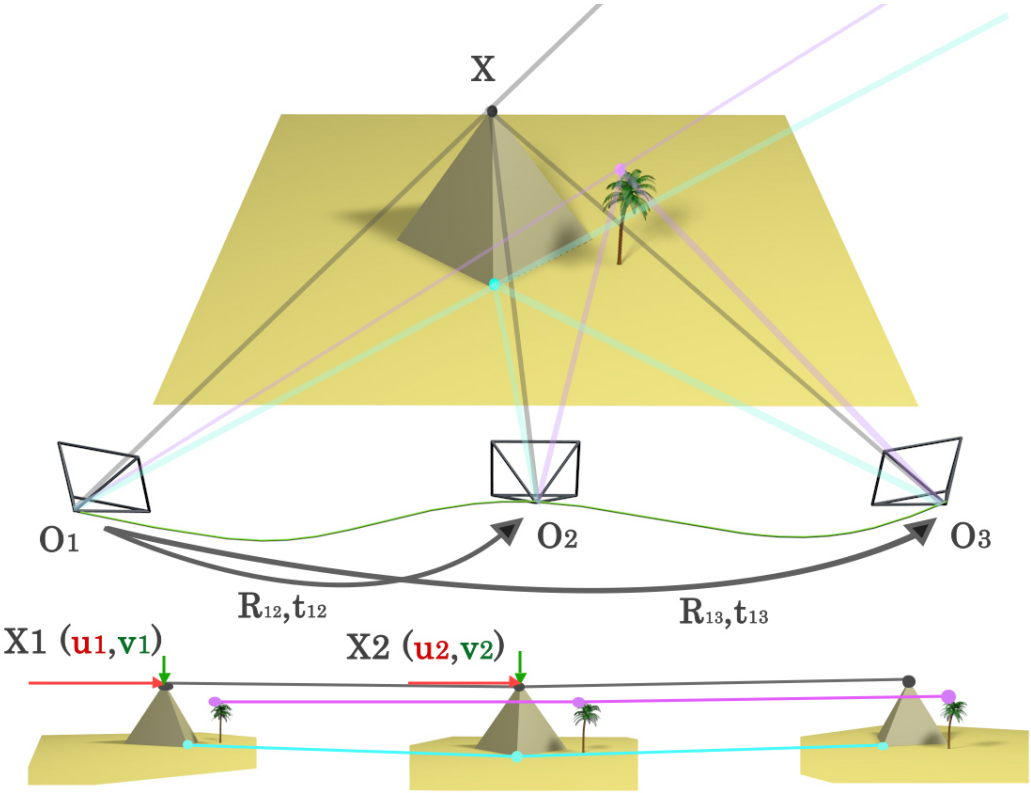

Figura 2.1: Trayectoria monocular (en verde). Se muestra la localización en los fotogramas 1 y 2 del punto 3D denominado X correspondiente al pico de la pirámide proyectado con unas coordenadas (en píxeles) u y v. En gris, morado y cian los haces de luz y sus intersecciones en algunos puntos de interés.

El modelo representado en la figura 2.1 ilustra el sistema a resolver de forma sintetizada, siendo las incógnitas a despejar:

- Las sucesivas matrices de rotación R y vectores de desplazamiento t necesarios para localizar la cámara con respecto a su posición anterior entre los "i" fotogramas analizados. R y t suponen 5 incógnitas: 3 grados de orientación y dos de posición (altura y acimut con respecto a la anterior posición de la cámara).

- Las coordenadas 3D globales de los "j" puntos para los que se encuentran correspondencias (3 incógnitas por punto).

Por otro lado, el número de ecuaciones nos lo proporcionan:

– Los pares de coordenadas de la proyección en cada cámara que se pueden igualar al encontrar una correspondencia. En el caso de la figura 2.1 $(u1, v1)$ con $(u2, v2)$. Por lo tanto, cada punto 3D genera dos coordenadas en cada plano de proyección dando lugar a 4 ecuaciones (se tienen dos cámaras y por lo tanto 2 planos de proyección).

Tabulando la variación de los grados de libertad (GL) del sistema en función del número de puntos para los que se encuentra una correspondencia se obtiene la tabla 2.1.

Tabla 2.1: Monocular: grados de libertad del sistema en función del número de puntos (correspondencias).

nºPuntos	Incógnitas	EQ	GL
1	8 (5+3)	4	4
2	11 (5+6)	8	3
3	14	12	2
4	17	16	1
5	20	20	0

Se observa que, a partir de los 5 puntos, el sistema comienza a estar sobredeterminado y por lo tanto es resoluble, aunque con cierto error. Se realiza a continuación una explicación más concisa del proceso iterativo que se lleva a cabo en el BA para obtener la solución con el error mínimo.

Primero se hace una estimación inicial de las nuevas R y t (combinadas: T_0) a partir de la dirección y velocidad que lleva la cámara. Se proyectan los X_j puntos 3D del mapa sparse en la estimación inicial de la cámara tras su supuesto movimiento T_0 (π_0 : función de proyección de la cámara para la estimación del cambio de posición T_0). Se busca entonces la correspondencia $x_{0,j}$ de la proyección de dichos puntos 3D en el nuevo fotograma. Cuando se encuentran, se mide la distancia (en píxeles) entre donde se creía que iba a estar proyectado el punto X_j y donde se ha encontrado su correspondencia $x_{0,j}$ realmente. El conjunto de las distancias supone el error $e_{0,j}$ de nuestra estimación inicial (2.1). A partir de la primera estimación T_0 se prueba entonces con nuevas T_i hasta conseguir el menor error $e_{i,j}$ (2.2).

$$e_{i,j} = x^m_{i,j} - \pi^m_i \left(T_{i,\omega}, X_{\omega,j}\right) \tag{2.1}$$

$$T = \arg\min_T \sum_{i,j} \rho \left(\left(x^m_{i,j} - \pi^m_i \left(T_{i,\omega}, X_{\omega,j}\right) \right)^2 \right) \tag{2.2}$$

Donde “m” hace referencia a “monocular”, ω a “referencia global” y ρ es la función de influencia robusta de Huber.

2.2. ORBSLAM stereo

En un par estéreo se puede obtener la para estéreo. Se estima la diferencia entre un punto punto ORB en el la cámara izquierda y su correspondencia en la derecha. A esta distancia se le denomina “disparidad”. A continuación se triangula su profundidad Z como se muestra en la figura 2.2.

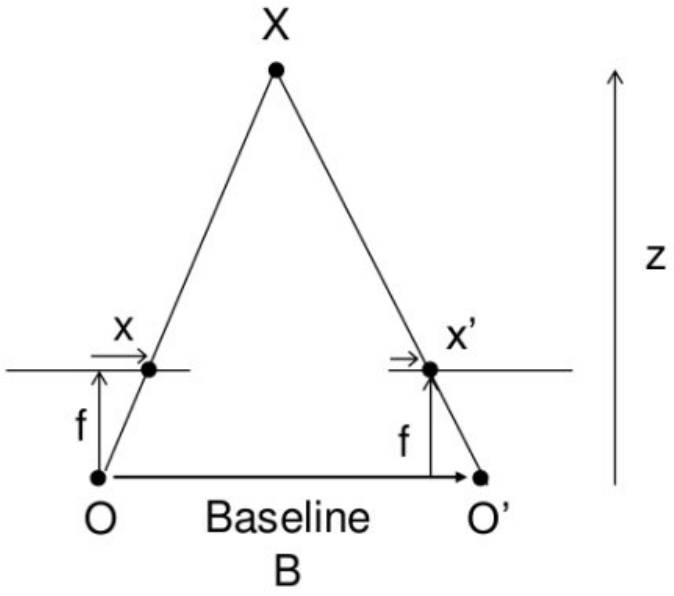

Figura 2.2: Obtención del la profundidad del estéreo: Donde la Baseline V, las distancias focales f y las coordenadas del punto en las proyecciones (x y x') son datos conocidos y $Z = \frac{Bf}{X-X'}$.

Esto permite la búsqueda de correspondencias de un punto visto por diferentes pares directamente en el espacio 3D (figura 2.3).

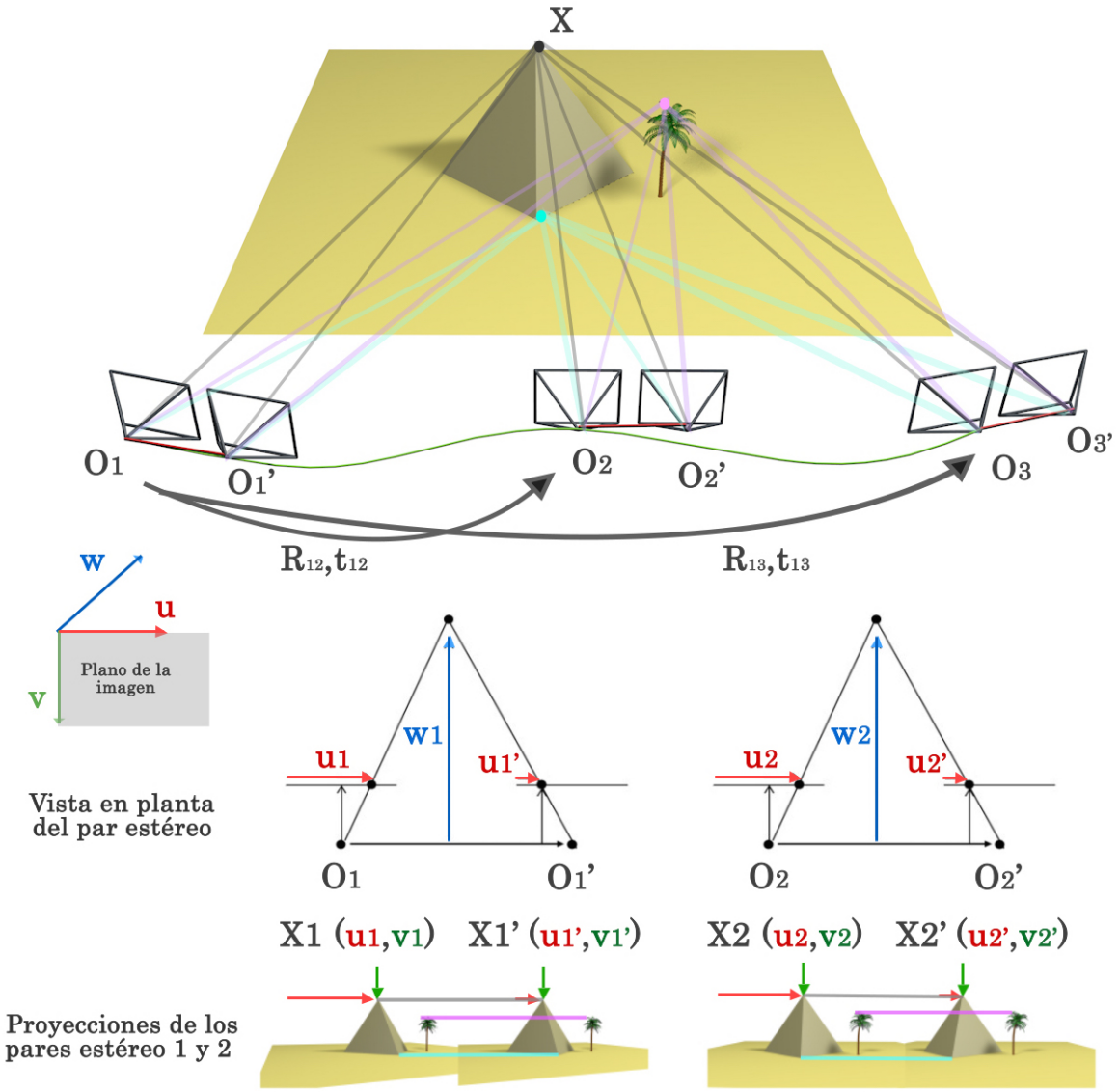

Figura 2.3: Trayectoria estéreo (en verde): La profundidad del punto 3D X se obtiene tanto en el fotograma uno como en el dos mediante la triangulación del par estéreo correspondiente a cada fotograma. En gris, morado y cian los haces de luz y sus intersecciones en algunos puntos de interés.

La figura 2.3 ayuda a sintetizar, como se ha hecho anteriormente en monocular, el problema a resolver: la obtención del mapa y la localización de la cámara con respecto al mapa. El estéreo puede triangular el 3D de un punto en un solo fotograma (figura 2.2), lo que modifica la relación de incógnitas y ecuaciones que se obtienen en función del número de puntos. Las incógnitas serían:

– Las matrices R y t (juntas: T) como en el caso monocular, pero en este caso, se añade la correspondiente a la escala (el módulo del vector desplazamiento es arbitrario en el caso monocular pero no lo es en estéreo) puesto que, al conocer la distancia entre las cámaras del par estéreo (línea base o baseline), se puede

conocer la magnitud de las distancias obtenidas. Esto se traduce en 6 incógnitas a resolver.

Por otro lado, el número de ecuaciones viene dado por la localización del punto en el espacio tridimensional. 3 coordenadas espaciales que se traducen en 3 ecuaciones por punto estéreo.

Como en el caso monocular, se ha tabulado la relación entre número de puntos obtenidos y los grados de libertad del problema en la tabla (2.2).

Tabla 2.2: Estéreo: grados de libertad del sistema estéreo en función del número de puntos obtenidos (correspondencias estéreo).

nºPuntos	Incógnitas	EQ	GL
1	6	3	3
2	6	6	0

Con 2 puntos no alineados el sistema ya podría ser resoluble según argumento de conteo, sin embargo, hacen falta 3 puntos no alineados en 3D para que el sistema tenga solución única. El proceso iterativo es muy similar al explicado en monocular, pero se considera la proyección estéreo. Es decir, en monocular $x^m_{i,j} \in \mathbb{R}^2$ mientras que en stereo $x^s_{i,j} \in \mathbb{R}^3$.

$$e_{i,j} = x^s_{i,j} - \pi^s_i \left(T_{i,\omega}, X_{\omega,j}\right) \tag{2.3}$$

$$T = \arg\min_T \sum_{i,j} \rho \left(\left(x^s_{i,j} - \pi^s_i \left(T_{i,\omega}, X_{\omega,j}\right)\right)^2 \right) \tag{2.4}$$

Donde "s" hace referencia a "stereo", ω a "referencia global" y ρ es la función de influencia robusta de Huber.

2.3. Comparación entre SLAM monocular y stereo

En el estéreo, la base entre las 2 cámaras tiene que ser suficientemente grande para obtener buen paralaje y poder triangular los puntos en el espacio. Esto supone un inconveniente para trabajar en espacios reducidos como es el caso de las endoscopias. Por otro lado, el estéreo presenta la siguientes ventajas en el análisis de secuencias de endoscopias:

– **Mejor respuesta ante superficies que sufran deformaciones**: uno de los principales inconvenientes de trabajar en monocular es que, entre una posición de la cámara y la siguiente, la escena, si no es rígida, ha podido cambiar. Esto

hace que se interprete el movimiento no rígido como paralaje obteniéndose, por lo tanto, profundidades erróneas. Este suceso no es poco habitual en el interior del cuerpo humano, puesto que contiene muchos tejidos semi-rígidos y deformables. En el caso del par estéreo, una deformación de una superficie tiene menos impacto al no depender de otros fotogramas para la generación de puntos. Esta ventaja permitiría, también, una segmentación robusta del mapa, es decir: diferenciar las zonas que realmente se han deformado de las que no.

- **Robustez en la inicialización**: la inicialización del mapa de puntos es un proceso bastante complejo y puede llegar a ser un factor limitante en la viabilidad del programa en el caso monocular, sin embargo, en el caso estéreo se puede inicializar el mapa de forma mucho más robusta. Un solo par de imágenes estéreo ya puede generar una región de mapa inicial que sirva de punto de partida para la inicialización, logrando a su vez ampliar el mapa con menos puntos que en el caso monocular.

- **Generación de mapa a escala real**: si se conoce la distancia que separa las dos cámaras se conoce también la distancia entre los puntos que se obtiene de las triangulaciones (figura 2.2). Esto se traduce en que el mapa que se genera durante una inspección en estéreo contiene distancias conocidas y mensurables entre sus puntos.

2.4. Re-localización y cerrado de bucles

2.4.1. Concepto de KeyFrame

El concepto de KeyFrame es necesario para entender los procesos re-localización y cerrado de bucles. Los KeyFrames o fotogramas clave son aquellos fotogramas que tienen la suficiente separación entre ellos como para definir la escena. Las fronteras (por lo tanto los puntos ORB) de su campo de visión se solapan (figura 2.4).

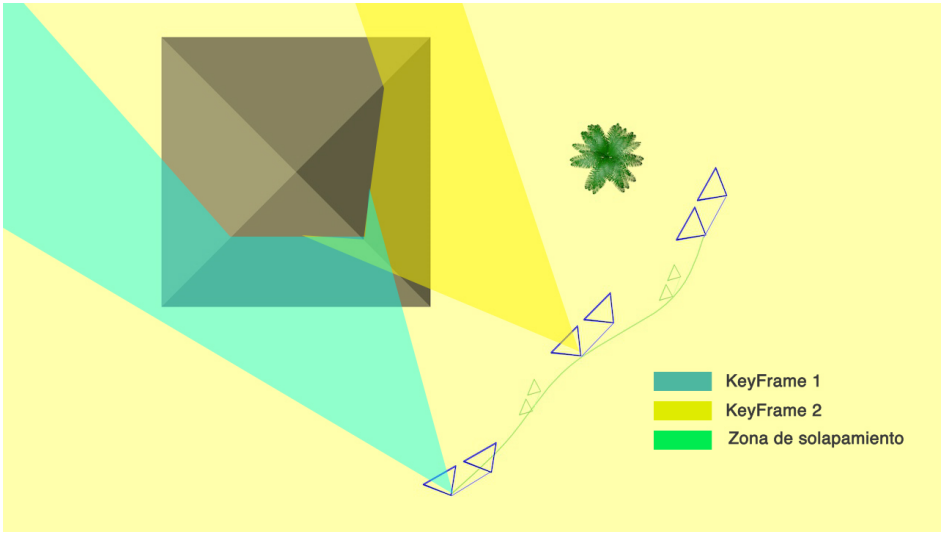

Figura 2.4: Vista en planta de una escena. Verde: la trayectoria de la cámara y los fotogramas normales (pares estéreo). Azul oscuro: de mayor tamaño los KeyFrames (pares estéreo también). Se muestran también dos cortes coplanarios de los campos de visión de los dos primeros keyframes (correspondientes a sus cámaras izquierdas).

2.4.2. Re-localización

En ORBSLAM se puede realizar la comparación de un fotograma con todos los KeyFrames ya creados (bolsa de palabras) para encontrar el que más se le parece. Se intenta entonces realizar un BA con el KeyFrame más similar, la región de mapa que observa dicho KeyFrame y el fotograma actual. Esto abre la posibilidad de re-localizar la cámara si el sistema se ha perdido, pero solo si la cámara está por una zona en la que ya se había generado mapa sparse.

2.4.3. Cerrado de bucles

La calidad de búsqueda de correspondencias está limitada por factores como la cantidad finita de píxeles (discretización de una realidad contínua) y la deriva en la escala. Estos factores inducen a un error que se puede ir acumulando obteniendo al final un mapa distorsionado y poco fiel a la realidad. Al guiarse la cámara con respecto al mapa se obtiene también una trayectoria errónea. Para corregir la trayectoria y el mapa en ORB-SLAM se ha añadido una función de detección y corrección de bucles. Cuando la cámara detecta que ha vuelto a una posición en la que ya ha estado y ha generado mapa (es decir, que ha cerrado un bucle) se re-ajusta el mapa definido por el bucle mediante una interpolación de la trayectoria (figura 2.5) logrando así una corrección que elimina toda la deriva acumulada.

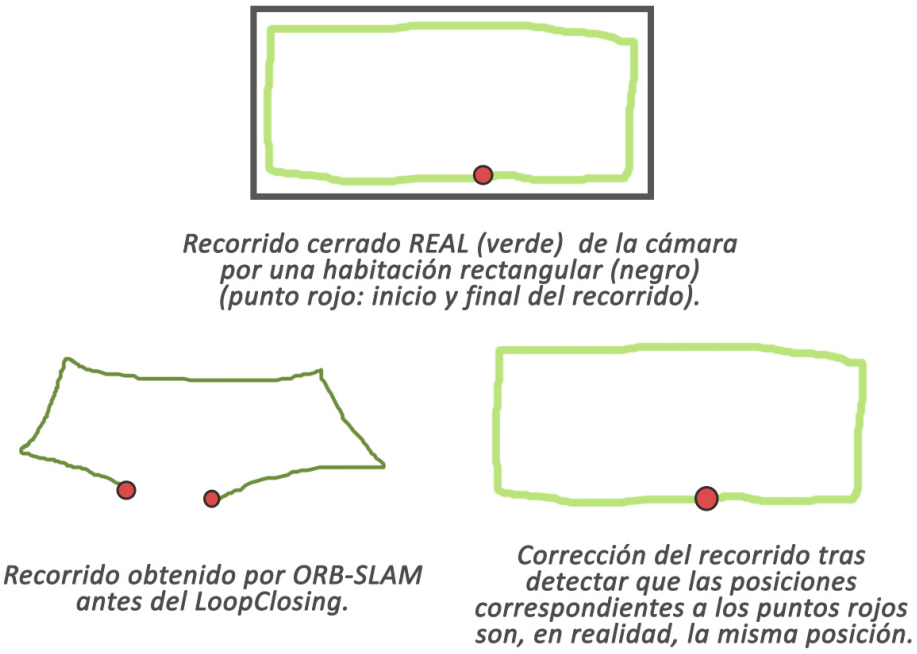

Figura 2.5: Representación del proceso de Loop Closing

Capítulo 3

Densificación en ORBSLAM

3.1. Densificación de un par estéreo: mapa denso local

En esta sección se explican los principios teóricos y el proceso de obtención de una observación densa a partir de un solo par estéreo.

3.1.1. Geometría epipolar y concepto de disparidad

Un punto en el espacio 3D visto por un par estéreo define un plano junto con los dos centros ópticos de las cámaras izquierda y derecha. Los cortes de ese plano con los dos planos de proyección definen 2 líneas epipolares, una por cada plano de proyección. Si se busca en la cámara derecha la correspondencia de un punto proyectado en la cámara izquierda se sabe, al resolver la geometría epipolar (ecuación 3.1), que dicha correspondencia va a estar en la línea epipolar de la cámara derecha (figura 3.1).

$$x'Fx = 0 \tag{3.1}$$

Donde F es la matriz fundamental (3x3) y Fx define la línea epipolar en la que x' se puede encontrar.

La matriz F engloba la posición y rotación relativa entre las cámaras, además de sus parámetros intrínsecos. Se puede aplicar una rotación a ambas cámaras y la consiguiente trasformación a sus matrices de proyección (parámetros instrínsecos), de forma que ambos planos de proyección sean coplanarios y que F' defina lineas epipolares horizontales para todos los puntos. Este proceso se denomina rectificación.

La disparidad, recordemos, es la diferencia en píxeles entre la proyección de un punto en cámara izquierda y la derecha del par estéreo. Trabajar con imágenes rectificadas permite que la búsqueda de disparidades se realice solo una la horizontal lo que disminuye notablemente el costo computacional.

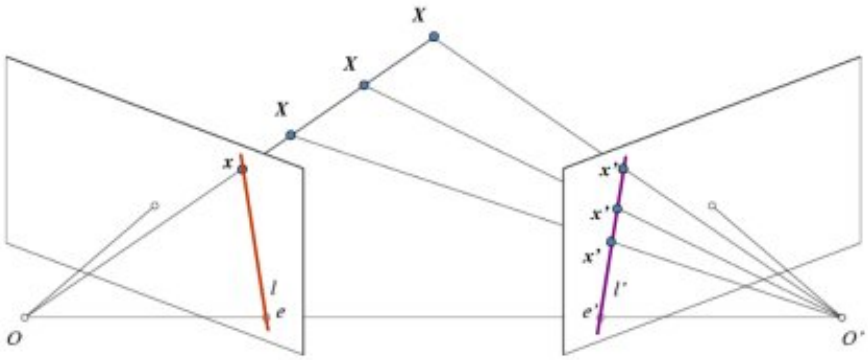

Figura 3.1: Par sin rectificar: plano epipolar y líneas epipolares para un punto X 3D y posibles posiciónes de su correspondencia en la cámara derecha x'.

En la figura 2.2 la disparidad correspondería a la distancia x-x' (en píxeles), de forma que, cuanto mayor sea la disparidad, menor será la profundidad del punto y, cuanto menor sea la disparidad, a más distancia se encontrará dicho punto con respecto a las cámaras. Una disparidad muy pequeña, dada la resolución limitada del sensor, conlleva un mayor error en el cálculo de la profundidad.

3.1.2. Rectificación de las secuencias de la base de datos de Hamlyn

Las imágenes estéreo obtenidas de la base de datos del Hamlyn Centre requieren de una rectificación y, además, presentan distorsión. La distorsión produce líneas epipolares curvas por lo que es necesario eliminarla. Se realiza un mapa de eliminación de la distorsión y de rectificado con interpolación lineal utilizando función StereoRectify (interna de la librería de OpenCV). Este método computa las matrices de rotación, que hay que aplicar a cada cámara, y las matrices de proyección de ambas cámaras en sus nuevas coordenadas, para su rectificación (explicación completa del proceso en el Bradski [6], pág. 728). En la figura 3.2 se ilustra el proceso de eliminación de la distorsión y posterior rectificación y, en las figura 3.3, el resultado de la aplicación de dicho proceso a un par de imágenes estéreo de uno de los archivos de la base de datos del Hamlyn Centre.

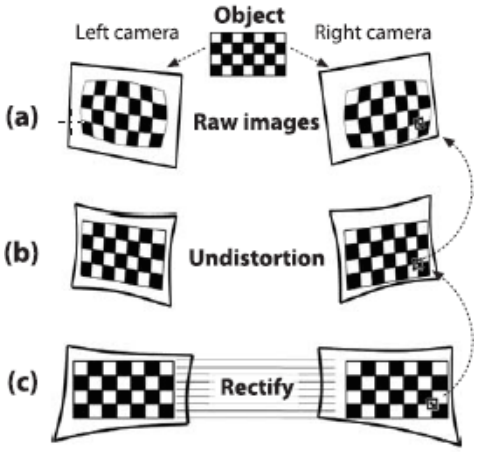

Figura 3.2: Proceso de eliminación de distorsión y de rectificación: (a)Imágenes en bruto, (b)Eliminación de la distorsión y (c)Rectificación (ilustración del libro Bradski [6]).

3.1.3. Generación del mapa de disparidades para un par estéreo

Una vez rectificado el par estéreo se puede proceder a la búsqueda de disparidades. Se ha utilizado la función "compute" de la clase StereoBM (librería OpenCV [7]) que utiliza emparejamientos de "ventanas" o regiones entre las imágenes de un par estéreo rectificado para obtener un mapa de disparidades. Se pueden modificar diversos parámetros en la generación del mapa de disparidades. Los dos parámetros más importantes son:

- **numDisparities**: hace referencia al rango de búsqueda de las disparidades. Para cada píxel el algoritmo buscará la mejor disparidad dentro de un rango que va desde "0" (valor por defecto, modificable) hasta el número con el que se configure "numDisparities". Aunque lo normal en escenas tan pequeñas y cercanas a la cámara sería que no existiesen puntos demasiado alejados, de existir algún punto lejano este arrastrarían un gran error. Los puntos excesivamente cercanos también suelen ser erróneos puesto que muy rara vez la cámara se acerca en exceso a ninguna superficie y, aunque se acercase, la sobre-exposición y los reflejos causados por la cercanía del foco darían lugar de nuevo a correspondencias erróneas. Por ello se ha aplicado un límite inferior de 10 píxeles y uno superior de 100 píxeles de disparidad.

- **blockSize**: Es el tamaño de los bloques o "ventanas" con los que se van a buscar las correspondencias. Cuanto mayor sea el tamaño, más liso y regular será el mapa y, cuanto menor blockSize, más detallado pero habrá mayor probabilidad de que el algoritmo encuentre correspondencias erróneas. Se han probado diversos

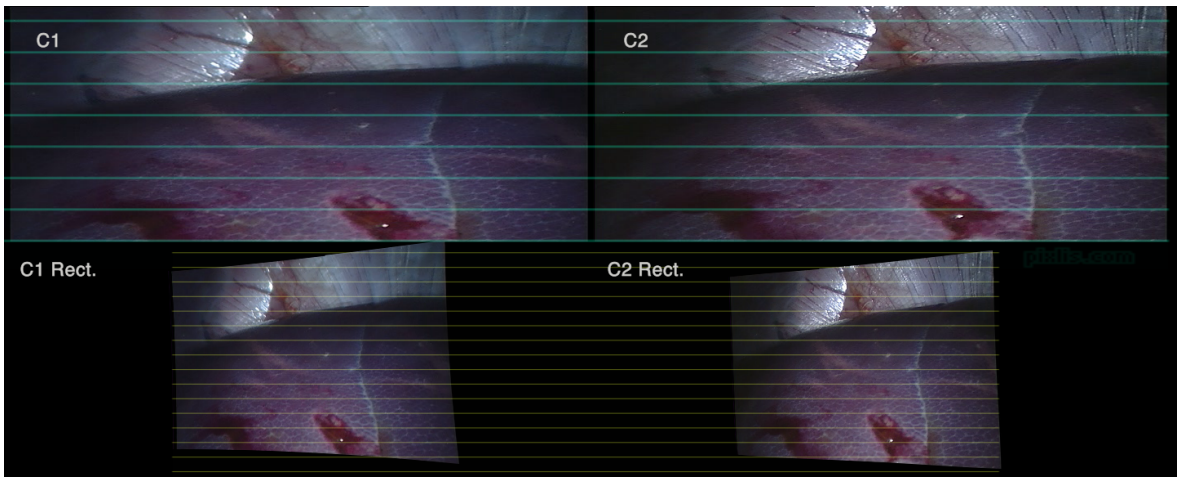

Figura 3.3: Par estéreo de una secuencia de la base de datos Hamlyn antes y después de la rectificación y la eliminación de la distorsión. Se puede apreciar cómo tras la rectificación, se encuentran correspondencias siempre en la misma horizontal (C1 y C2 cámaras izquierda y derecha respectivamente).

valores intentando lograr un compromiso entre regularidad del mapa y detalle. Al final se ha llegado a la conclusión de que el valor que mejor equilibrio lograba era el de 17 píxeles. Por ilustrar el impacto de valores excesivamente bajos (5 píxeles, el mínimo) o altos (45 píxeles) se muestra una comparativa en la figura (3.4).

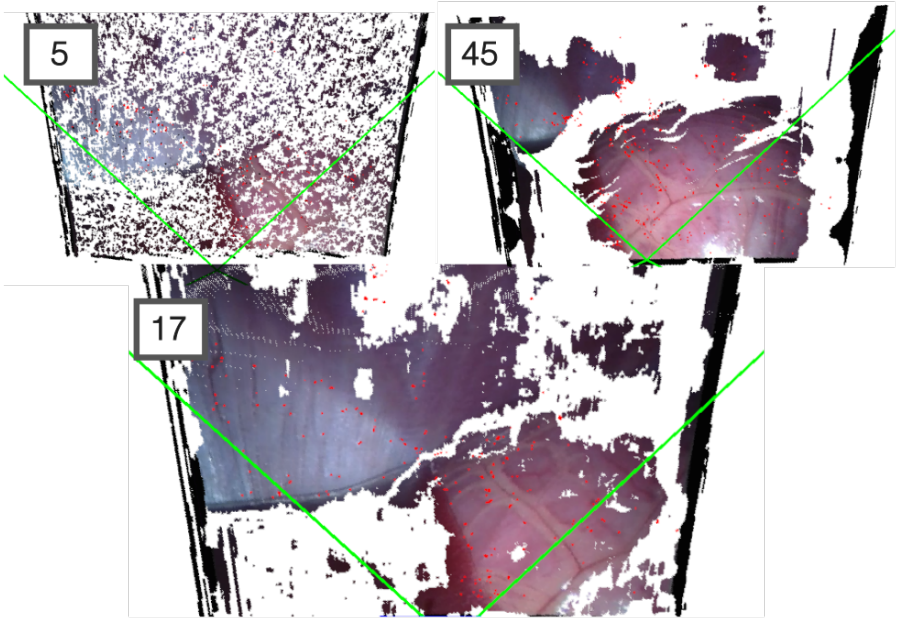

Figura 3.4: Comparación de mapa denso con diferentes valores del parámetro blockSize. Se observan demasiados puntos dispersos para el valor de 5 píxeles (valor mínimo) y demasiados huecos con valores demasiado altos como 45 píxeles. El valor de 17 consigue bastante densidad y suficiente detalle.

3.1.4. Generación de los puntos 3D obtenidos a partir del mapa de disparidades.

El mapa de disparidades proporciona la profundidad Z de un punto de la imagen izquierda. Multiplicando sus coordenadas en píxeles por la inversa de la matriz de proyección P de la cámara izquierda se obtienen sus otras dos coordenadas 3D X e Y con respecto al centro óptico de la cámara izquierda. Se almacenan las coordenadas de los puntos densos junto con sus correspondientes valores de color RGB y se representan en el espacio 3D obteniendo resultados como los de la anterior figura 3.4.

Se aprecia un marco negro que encuadra la parte densificada. Ese marco negro es resultado de los emparejamientos de los puntos correspondientes al borde de la imagen. La buena iluminación en el centro de la imagen va dando paso gradualmente a una zona oscura en los bordes. Se genera, por lo tanto, un gradiente de grises por todo el contorno de la imagen. Este gradiente aumenta mucho la posibilidad de encontrar disparidades por las regiones limítrofes con los bordes del fotograma, lo que se traduce en el marco negro/gris una vez llevado al 3D.

Se elimina este problema añadiendo al mapa solamente aquellas disparidades que entren dentro de la zona bien iluminada, haciendo un recorte que engloba la zona central de la imagen obteniendo un mapa local denso más limpio (imagen 3.5).

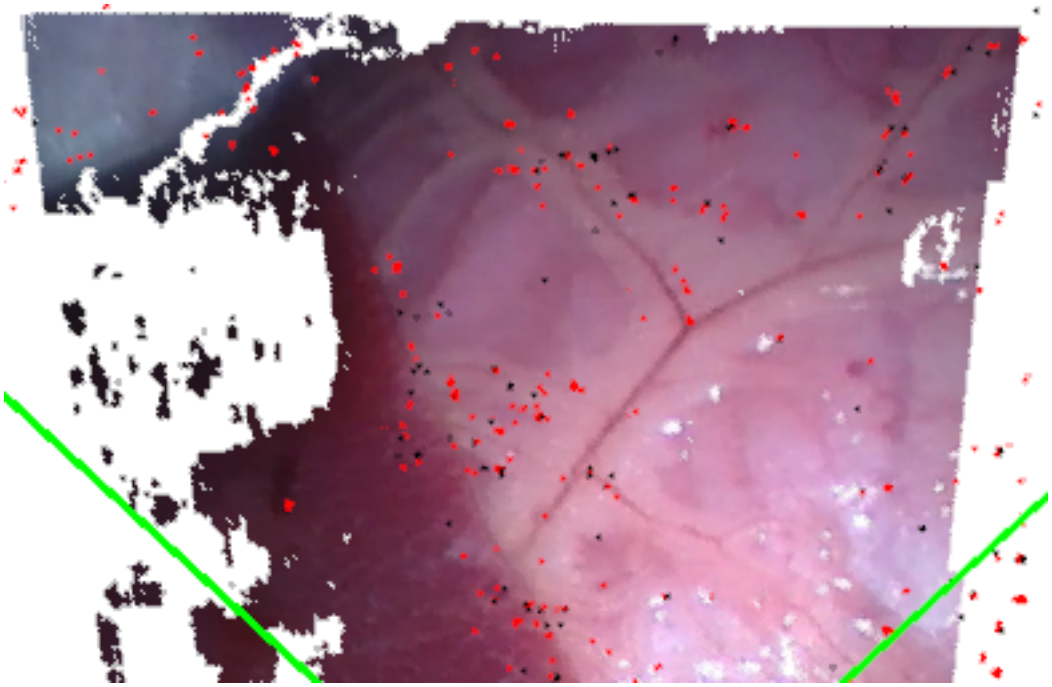

Figura 3.5: Mapa denso local tras el recorte del marco. Los puntos negros y rojos superpuestos corresponden a puntos del mapa sparse.

3.2. Mapa denso global

En esta sección se expone el método utilizado en el alineamiento de los diversos mapas locales para la obtención de un mapa denso global de la escena observada.

3.2.1. Alineación de los mapas densos locales

Como se ha explicado anteriormente, los KeyFrames son suficientes para definir la escena por lo que, si se quiere realizar un mapa denso global, es suficiente con obtener mapas densos locales en cada KeyFrame y alinear dichos mapas. En ORBSLAM, los KeyFrames se guardan con una serie de datos, entre ellos su posición con respecto a la referencia global. Esto permite transformar las coordenadas locales de los puntos densos obtenidos para un KeyFrame en coordenadas globales y, por lo tanto, ir alineando los diferentes mapas densos locales formando uno global.

3.2.2. Rechazo de los puntos densos duplicados

Cuando las zonas de solapamiento entre dos KeyFrames son demasiado extensas o se vuelve a pasar por una zona ya explorada del entorno, se producen puntos densos duplicados. Una gran cantidad de puntos duplicados supone un costo computacional muy alto además de no aportar datos desconocidos.

Este problema se soluciona mediante la proyección en el nuevo KeyFrame de los puntos densos ya existentes. Se genera una matriz de ceros de igual tamaño que la resolución del KeyFrame actual y se cambia el valor de “0” a “1” a aquellos píxeles de la matriz de ceros en los que se haya proyectado un punto 3D ya existente. Cuando se va a añadir un nuevo punto 3D, obtenido del mapa de disparidades del KeyFrame actual, se comprueba que en las mismas coordenadas de la matriz de ceros no haya un “1” y, solo entonces, se añade el nuevo punto 3D. De esta forma, se evitan duplicaciones, se ahorra en procesos de cómputo para la representación 3D y se mejora la textura general homogeneizándola.

3.3. Conclusiones

El mapa sparse se crea para orientar la cámara y facilitar la obtención de su trayectoria, pero no da información fiable sobre el posicionamiento de la cámara con respecto a las superficies densas que la rodean. Una densificación de estas características facilitaría el evitar posibles colisiones, la identificación y medida de diferentes partes de la escena y, en definitiva, una interacción más precisa con la escena a tiempo real.

Capítulo 4

Resultados experimentales

4.1. Organización y acondicionamiento de los datos del Hamlyn Centre

La generación de un sistema SLAM estéreo en ORBSLAM necesita, como inputs, los tres elementos enumerados a continuación:

1. Los fotogramas ya rectificados correspondientes a las cámaras izquierda y derecha.

2. Una lista de parámetros en formato ".yaml". El archivo con dicha extensión incluye parámetros referentes a:

   - Parámetros de la cámara:
     - Intrínsecos: distancias focales fx, fy y coordenadas del centro óptico cx,cy.
     - De calibración y distorsión: todos con valor 0.0 (nulo) puesto que se han rectificado los pares estéreo, es por lo tanto un modelo de cámara proyectiva.
   - Parámetros de rectificación: no necesarios para la generación del sistema SLAM, pero sí para la rectificación previa de los pares estéreo.
   - Parámetros de ORB: parámetros modificables como el número de puntos ORB por imagen, niveles del factor de escala, el paso entre ellos y valores iniciales y mínimo del Fast threshold.
   - Parámetros de visualización: los parámetros con los que se pueden modificar los tamaños de representación de los KeyFrames, la trayectoria, la cámara y la posición relativa del observador.

Se puede encontrar un ejemplo de archivo ".yaml" en el anexo A.

3. Una lista en un archivo ".txt" que contenga los "timestamps" o listado numérico identificativo de los fotogramas.

La base de datos del Hamlyn Centre de Londres incluye, entre otras cosas, diversas escenas de laparoscopias (en cerdos) grabadas en estéreo. Las escenas se presentan en un archivo de vídeo ".avi" acompañado de tres archivos de calibración ".txt". El archivo de vídeo contiene las grabaciones tanto de la cámara izquierda como de la derecha reproducidas simultáneamente una al lado de la otra. Los tres archivos de calibración consisten en:

– Matrices de parámetros intrínsecos y coeficientes de distorsión de las cámaras izquierda y derecha (cámaras C1 y C2 respectivamente).

– Matriz de rotación R y vector de posición t que relaciona la posición y la orientación relativa entre las dos cámaras: la derecha con respecto a la izquierda.

Se ha generado un programa denominado "StereoVidsToFrames" para obtener los elementos de entrada necesarios para ORBSLAM a partir de los que incluye la base de dayos del Hamlyn Centre. Las salidas de "StereoVidsToFrames" supondrán las entradas de "stereo_hamlyn", programa creado para rectificar las imágenes y llamar a la función de trackeo de ORBSLAM. "stereo_hamlyn" llama a dicha función, entregándole tanto las imágenes ya separadas y rectificadas, como la lista de timestamps. El orden de uso de los programas y los inputs y outputs que necesitan y generan quedan recogidos en la siguiente figura (4.1).

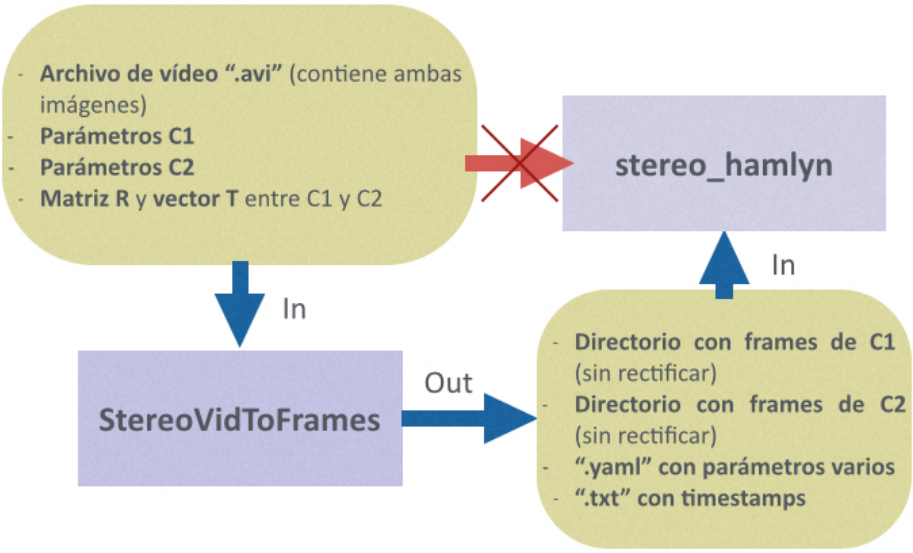

Figura 4.1: Esquema de programas (azul) y entradas/salidas (amarillo).

4.2. Sintonía de ORBSLAM

Se ha realizado, a continuación, una sintonización de ciertos parámetros de ORB-SLAM2 en la línea de los ajustes que realizó Iñigo Cirauqui para el análisis de endoscopias en monocular (la descripción completa de los parámetros se puede encontrar en el apartado 4.2 de su TFG [2]). Quedan tabulados los cambios en 4.1.

Tabla 4.1: Tabulación de los parámetros modificados para la sintonización del programa.

Descripción	Source	Línea	Antes	Ahora	Ud.
Tracking					
Área de búsqueda	ORBmatcher.cc	130	2.5/3.0	4.0/4.5	píxel
Invarianza a la escala	Frame.cc	163	1.0	0.9	-
Gestor del mapa, creación de puntos					
Distancia entre keyframes	LocalMapping.cc	259	0.01	0.05	-
Comparación ORB	ORBMatcher.cc	38	50	45	bit
Settings					
Niveles de escala	settings en el ".yaml"	-	8	6	-
Factor entre niveles de escala	settings en el ".yaml"	-	1.2	1.1	-
Umbral de oFAST	settings en el ".yaml"	-	20	24	-
KeyPoints deseados	settings en el ".yaml"	-	1000	1200	-

Se explican a continuación los cambios aplicados a los parámetros de la tabla anterior:

- **Área de búsqueda**: el área al rededor de la anterior localización de un punto ORB en la que se va a volver a buscar en el siguiente fotograma. Se aumenta, puesto que al trabajar en entornos no rígidos, las deformaciones pueden desplazar los puntos ORB, haciendo que aparezcan alrededor de áreas mayores.

- **Invarianza a la escala**: disminuir la invarianza a la escala implica ampliar el rango de búsqueda de puntos en profundidad pero solo hacia la cámara. Esto propiciará la búsqueda de puntos más cercanos, abundantes en este tipo de escenas.

- **Distancia entre Keyframes**: se aumenta la distancia entre Keyframes para obtener un mayor paralaje entre ellos. Además, al disponer de un par estéreo, no es necesaria tanta densidad lineal de KeyFrames.

- **Comparación ORB**: disminuir la Comparación ORB implica acortar la distancia de Hamming necesaria entre los descriptores para su emparejamiento es decir, hacer emparejamientos más estrictos. Aunque se emparejen menos puntos, estos serán más fiables.

– **Niveles de escala**: el número de cambios de escala en la búsqueda de correspondencias. Se disminuye al encontrarnos en un entorno reducido donde no existe mucha variedad entre la escala y la cámara tiene pocos desplazamientos en Z.

– **Factor entre niveles de escala**: se reduce el paso entre niveles de escala para moverse por un rango menor de cambio de escala por la misma razón que en el punto anterior.

– **Umbral de oFAST y KeyPoints deseados**: se aumenta el número de KeyPoints deseados y el umbral de oFAST para compensar el aumento de la restricción realizada al modificar "Comparación ORB". En definitiva, se crean más puntos en bruto pero se aplican filtros más duros para su emparejamiento.

También se ha modificado también un parámetro de la sección de "Viewer Paramters" en los archivos "hamlyn.yaml". Se le ha dado al tamaño de representación de la cámara en el visualizador ("Viewer.CameraSize") la medida de la distancia entre las cámaras del par estéreo. Esto permite tener cierta idea del tamaño que tiene el conjunto de las dos cámaras con respecto a la escena en observación.

4.3. Análisis de las secuencias y evaluación de ORBSLAM

Se han procesado un total de 20 secuencias de laparoscopias in-vivo y ex-vivo en cerdos con "StereoVidToFrames" y se ha evaluado ORBSLAM en todas ellas.

Cuando se evalúa un sistema SLAM suele haber una trayectoria de comparación, medida con otro tipo de sensores o métodos, que permiten una valoración cuantitativa de la precisión del sistema SLAM. En el caso de las endoscopias, es imposible obtener dichas trayectorias comparativas ya que, además de ser trayectorias únicas en cada intervención, también se analizan escenas únicas en cada instante de tiempo. Esto se debe a diversas causas: deformaciones por latidos, respiración, contracciones musculares, acción de las propias herramientas médicas, la fuerza del aire que se introduce a presión para generar la cavidad en las laparoscopias etc. La única valoración cuantitativa que se puede realizar es la del tiempo de cómputo del proceso SLAM. El resto de valoraciones serán cualitativas.

Se va a presentar, a continuación, el análisis cualitativo de 6 de las 20 secuencias ya procesadas. Se han elegido las 6 secuencias más representativas de las diferentes situaciones y casos que se dan dentro de la base de datos. Se analizarán en orden de mayor a menor dificultad basándose el orden en los siguientes factores:

- **Calidad de la grabación**: entendiendo como buena calidad de grabación aquella que no tenga movimientos demasiado bruscos ni veloces. Una grabación suave y lineal.

- **Calidad de la iluminación**: es indispensable para la búsqueda de correspondencias una iluminación fija con respecto a la posición y orientación de la cámara.

- **Rigidez de la escena**: tomando como positiva una escena con mayoría de elementos rígidos o quasi-rígidos y negativa aquella que tengan superficies que sufran deformaciones.

- **Intervención de instrumentos**: cuanto más movimiento exista ajeno a la escena, más dificultad presenta la secuencia para el sistema SLAM.

- **Textura de las superficies**: una superficie con mucha textura facilita la búsqueda de emparejamientos y, por lo tanto, se considerará como factor positivo.

Cabe mencionar que las texturas de los tejidos porcinos tienen más contraste que la de los tejidos humanos, lo que puede facilitar el emparejamiento de puntos.

Con respecto al criterio a la hora de valorar la respuesta del sistema SLAM se tendrán en cuenta los siguientes factores:

- **Continuidad durante el trackeo**: que el sistema no se pierda ni se tenga que re-localizar, valorando muy negativamente si se pierde definitivamente y se ha de resetear.

- **Detección y cerrado de bucles (LoopClosing)**: la detección y cerrado de bucles, aunque no tiene por qué darse, mejora generalmente la trayectoria obtenida de la cámara por lo que también se valora positivamente.

- **Trayectoria coherente**: que la trayectoria que representa la cámara tenga cierta coherencia con respecto a lo que se observa durante la grabación. Una trayectoria puede ser incoherente cuando el sistema tome como referencia puntos de superficies deformables y represente en la trayectoria el movimiento relativo de dicha superficie en vez de el de la cámara con respecto al conjunto de la escena o a las superficies rígidas, por ejemplo.

Se exponen a continuación las 6 secuencias. A la izquierda se presentan diversos fotogramas en blanco y negro correspondientes a la imagen obtenida por la cámara izquierda del par estéreo. En dichos fotogramas, representados en verde, se pueden apreciar los puntos de ORB detectados. A la derecha se muestra el mapa de puntos

sparse: en negro los puntos definitivos y, en rojo, los puntos en proceso creación. También se muestra la trayectoria y la posición actual de la cámara, en verde, así como la posición de los diversos KeyFrames, en azul.

4.3.1. Secuencia 1

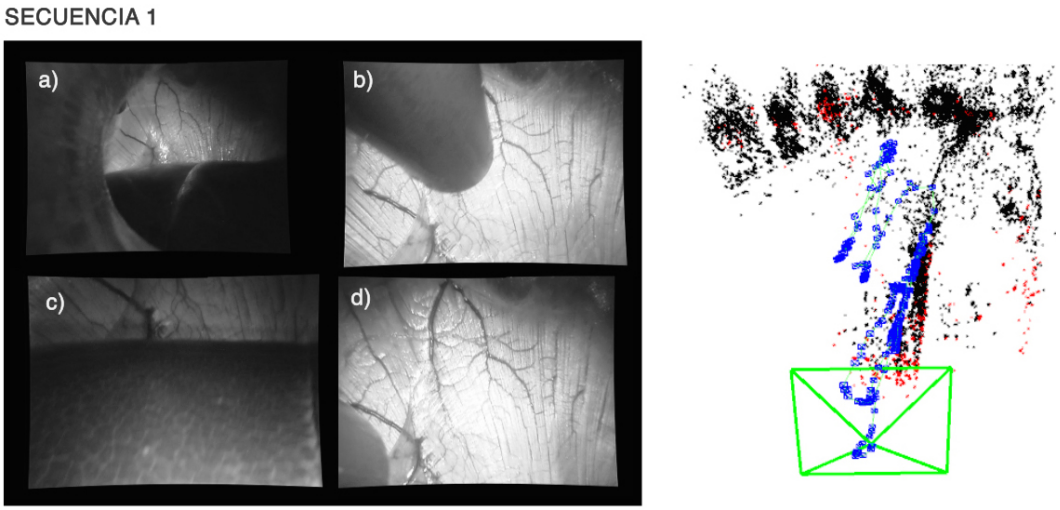

Figura 4.2: Secuencia 1.

– **Descripción**: Laparoscopia porcina in-vivo. Movimiento de cámara y de hígado causado por respiración. [8]

– **Elementos que dificultan el procesamiento de las imágenes**:

  - Calidad de grabación nefasta: Movimientos muy bruscos que se traducen en desenfoque de movimiento (c).
  - Foco de luz con movimiento muy variable: se desplaza repetidas veces por medio de la escena (b) y varía en exceso la iluminación. Este hecho, combinado con superficies muy reflectantes da lugar a zonas muy sobre-expuestas (d) o demasiado oscuras, puesto que no se da tiempo a adecuar el tiempo de exposición.
  - Entrada y salida de la cámara en la escena a través del tubo de inserción obligando a realizar un reset (a).

– **Elementos que facilitan el procesamiento de las imágenes**:

  - Escena con abundante textura que facilita el disparo del detector de puntos de interés.

– **Resultado**: Fallo en la relocalización, muchos resets necesarios, se muestra la mejor trayectoria obtenida a lo largo de toda la secuencia.

4.3.2. Secuencia 2

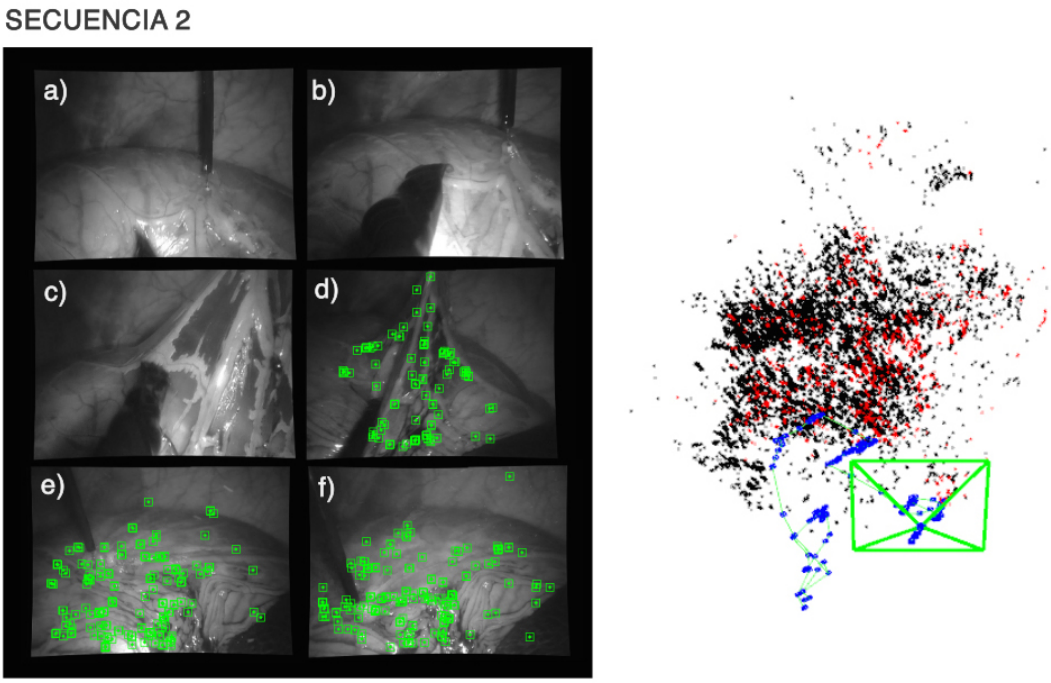

Figura 4.3: Secuencia 2.

– **Descripción**: Laparoscopia porcina in-vivo. Movimiento e interacción de herramientas con tejidos. [8]

– **Elementos que dificultan el procesamiento de las imágenes**:

  - Movimientos muy bruscos que facilitan la aparición de desenfoque de movimiento y por lo tanto la pérdida del tracking.
  - Gran interacción de las herramientas con la escena (b) llegando prácticamente modificarla por completo (c).
  - Escena con movimientos no rígidos causados por la respiración.

– **Elementos que facilitan el procesamiento de las imágenes**:

  - Escena con abundante textura.
  - Iluminación adecuada: luz fija con respecto a la cámara, correcta iluminación del fondo.

– **Resultado**: Gran dificultad de la secuencia. Se consigue alguna trayectoria con bastantes puntos, especialmente cuando la cámara se guía con puntos del fondo de la escena (e)(f). Además, si las herramientas se mueven lo suficiente, el sistema las ignora correctamente.

4.3.3. Secuencia 3

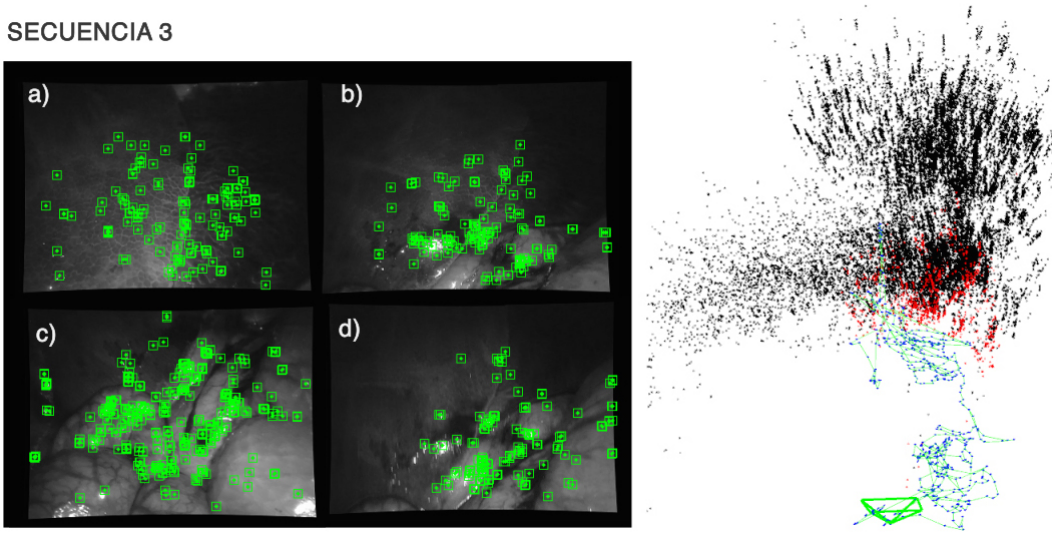

Figura 4.4: Secuencia 3.

– **Descripción**: Laparoscopia porcina in-vivo. Movimiento general de la cámara por el abdomen con movimiento del hígado debido a la respiración. [8]

– **Elementos que dificultan el procesamiento de las imágenes**:

  - Movimientos bruscos que facilitan la aparición de desenfoque de movimiento.
  - Escena con abundantes movimientos no rígidos causados por la respiración llegando a presentarse hasta 3 superficies diferentes con deformaciones y cambios de posición relativos entre ellas.

– **Elementos que facilitan el procesamiento de las imágenes**:

  - Escena con abundante textura. En este caso cabe destacar también la cantidad de contornos que sirven de soporte para una gran cantidad de puntos ORB (c).
  - Iluminación buena: luz fija con respecto a la cámara.

– **Resultado**: Una trayectoria larga y compleja con varios procesos de LoopClosing. En algunos momentos, se producen alteraciones poco coherentes en la trayectoria cuando hay un exceso de movimiento entre las superficies, pero el sistema no se pierde y puede continuar orientándose posteriormente.

4.3.4. Secuencia 4

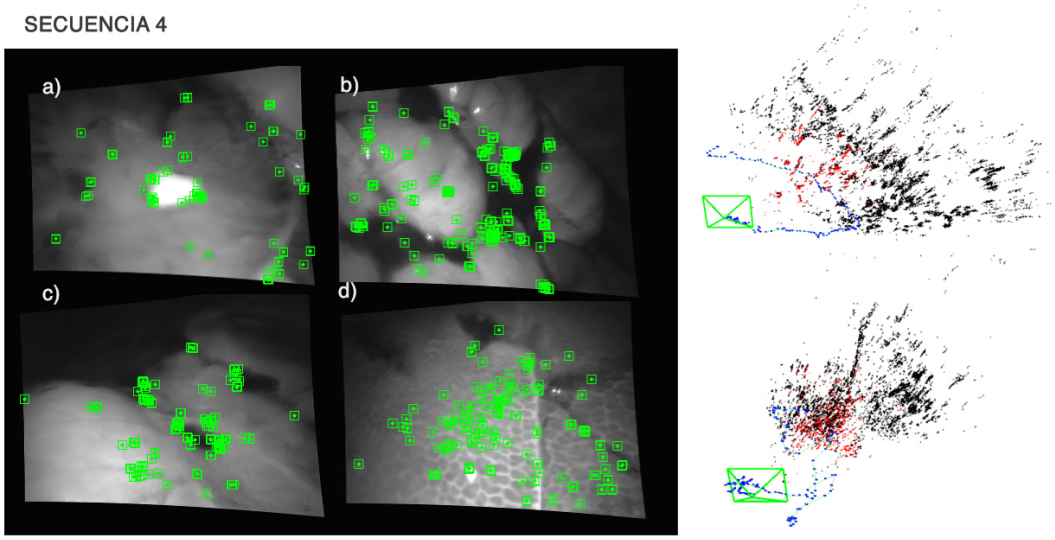

Figura 4.5: Secuencia 4.

– **Descripción**: Laparoscopia ex-vivo. Movimiento general de la cámara por el abdomen (estático). [8]

– **Elementos que dificultan el procesamiento de las imágenes**:

  - Abundante desenfoque ocasionado por la condensación de agua en la lente (a)(c).
  - Zonas de la escena con poca textura.
  - Abundantes movimientos angulares a mitad de la secuencia.

– **Elementos que facilitan el procesamiento de las imágenes**:

  - Calidad de grabación muy buena: un movimiento suave con poca aceleración.
  - Escena con abundante textura (d).
  - Iluminación muy favorable: luz fija con respecto a la cámara que, al moverse suavemente, da tiempo al ajuste del tiempo de exposición.
  - Escena prácticamente estática en su totalidad (ex-vivo).

– **Resultado**: Una trayectoria larga, compleja y coherente. En esta secuencia, dado la poca nitidez de algunos fotogramas (a), ORBSLAM demuestra su capacidad para localizarse con puntos o texturas desfavorables.

4.3.5. Secuencia 5

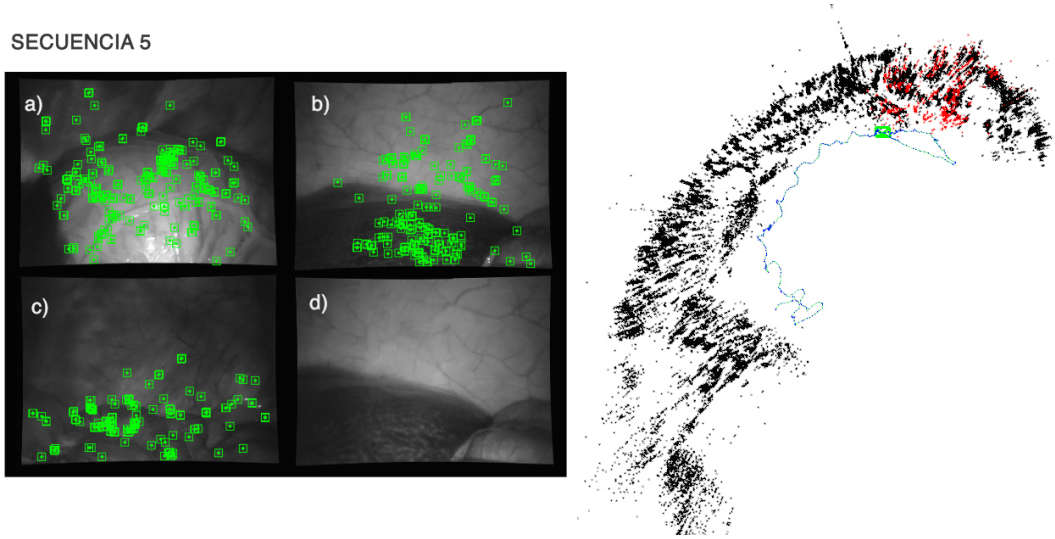

Figura 4.6: Secuencia 5.

– **Descripción**: Panoramización de abdomen porcino in-vivo. [8]

– **Elementos que dificultan el procesamiento de las imágenes**:

  - Movimientos, aunque muy pequeños, relativos entre superficies.

– **Elementos que facilitan el procesamiento de las imágenes**:

  - Calidad de grabación muy buena: un movimiento suave y lineal durante todo el tiempo.
  - Escena con abundante textura (a).
  - Iluminación muy favorable: foco fijo con respecto a la cámara que permite buena visualización del fondo de la escena.

– **Resultado**: Un excelente resultado. Una trayectoria larga y un mapa de puntos que refleja el movimiento panorámico realizado por la cámara. Trayectoria suave y coherente, sin saltos.

4.3.6. Secuencia 6

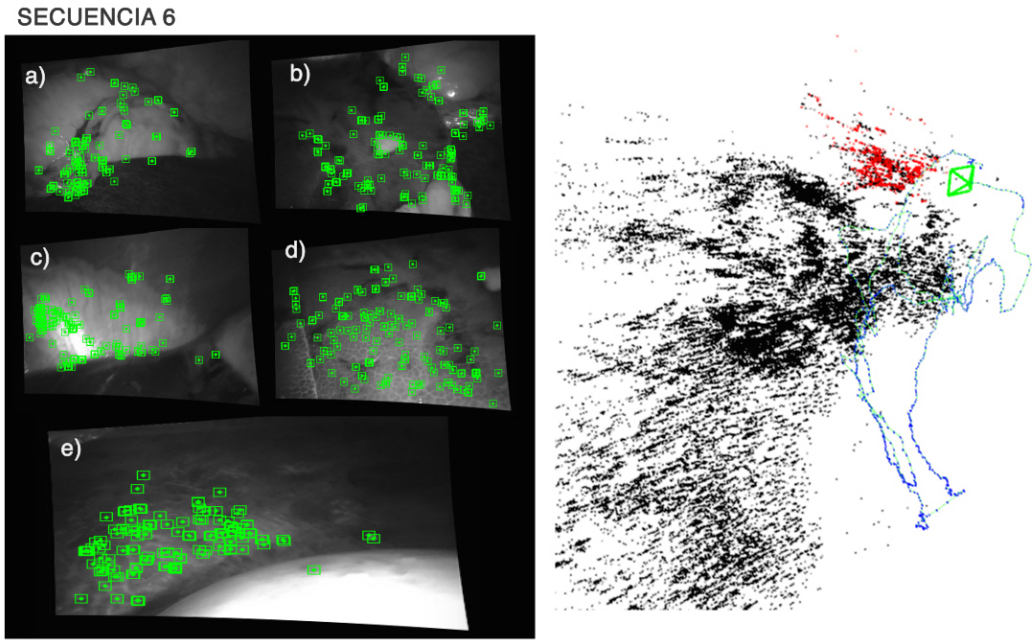

Figura 4.7: Secuencia 6.

– **Descripción**: Laparoscopia ex-vivo. Movimiento general de la cámara por el abdomen (estático). [8]

– **Elementos que facilitan el procesamiento de las imágenes**:

  - Calidad de grabación excelente.
  - Escena con abundante textura.
  - Iluminación muy favorable.
  - Escena rígida (ex-vivo).
  - Cerrado de bucles.

– **Resultado**: Un resultado modelo. Una extensa nube de puntos, una trayectoria larga y enrevesada pero con curvas suaves.

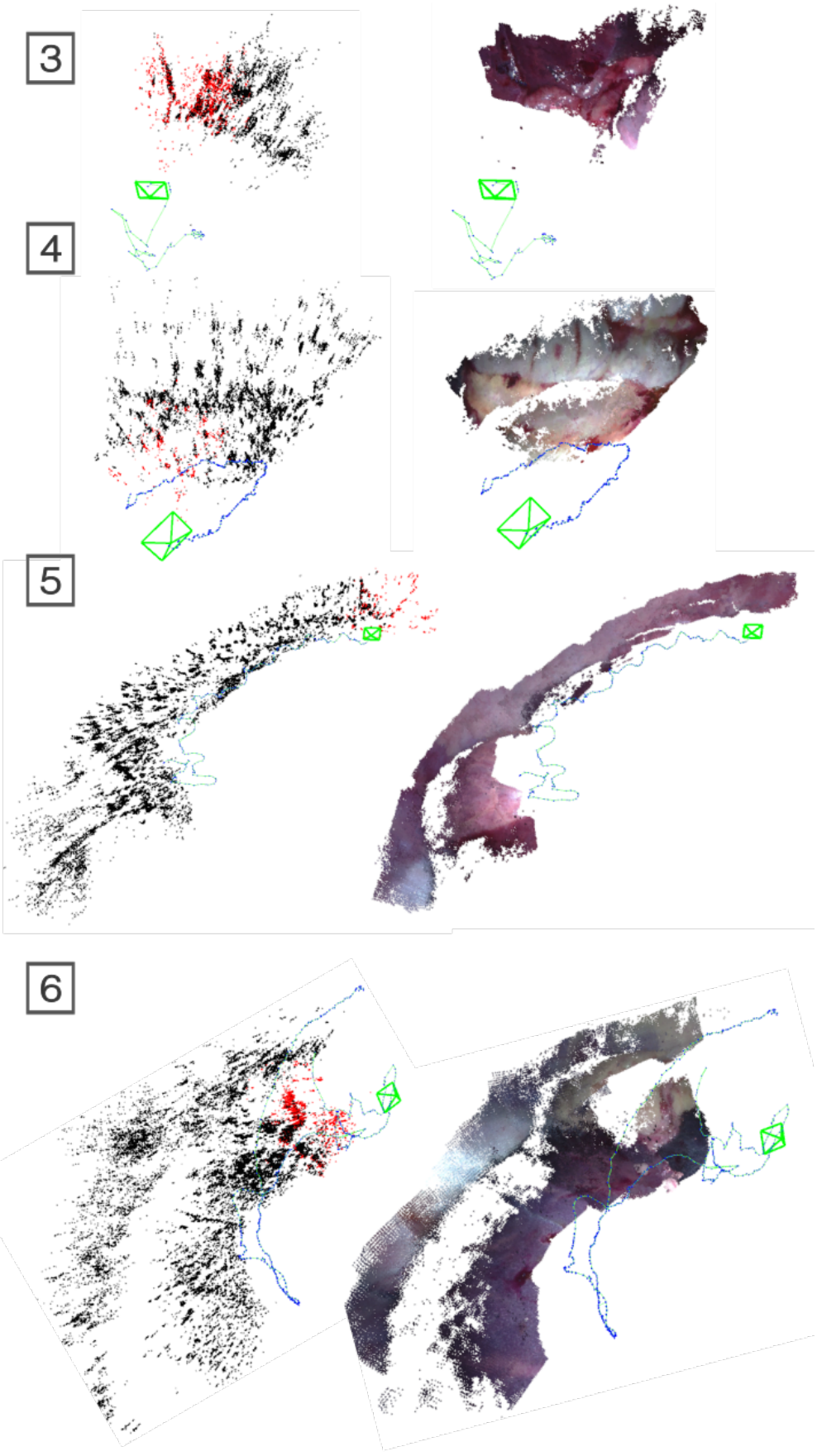

Figura 4.8: Comparaciones de los mapas sparse y densos de las secuencias 3,4,5 y 6.

4.4. Conclusiones

ORBSLAM estéreo en endoscopias presenta cierta dificultad para continuar el mapa o relocalizarse en las escenas grabadas con movimientos bruscos (conllevan desenfoque de movimiento). También, en algunas partes de la trayectoria, muestra tramos poco coherentes al orientarse respecto a superficies que presentan demasiadas deformaciones (secuencia 3). Por otro lado, ORBSLAM presenta un buen comportamiento en escenas con movimientos de cámara suaves y sin aceleraciones bruscas, que tienen una iluminación fija solidaria al movimiento de la cámara. También responde de forma adecuada ante pequeñas deformaciones. Siendo esta la primera aproximación del módulo estéreo en endoscopias con el programa sin modificar, se considera que ORBSLAM ha funcionado correctamente. Se realizan las siguientes propuestas de acuerdo a los resultados obtenidos durante este análisis:

- Potenciación de la generación de más puntos de ORB en el fondo de la escena, bajo la suposición de que, en los primeros planos, se van a realizar las intervenciones quirúrgicas y, por lo tanto, habrá más deformaciones y variaciones que en el fondo.

- Marcado de las herramientas con patrones reconocibles para facilitar su omisión en el proceso de trackeo.

- Realización de la segmentación de la escena: distinción de las regiones de la escena que se deforman de aquellas zonas que sean rígidas. Esto evitaría trayectorias incoherentes y supondría un primer paso para la identificación de los elementos deformables y la posterior generación de un mallado que facilite su seguimiento.

4.5. Tiempos de cómputo

4.5.1. Tablas

Se presentan las tablas con los datos sobre los tiempos de cómputo tanto del mapa sparse como del mapa denso. Los datos del mapa sparse de las secuencias en las que se requería realizar un reset (secuencias 1,2 y 3) se han calculado para trayectorias parciales. Con respecto al mapa denso, se presentan los datos obtenidos de fragmentos de las secuencias puesto que tras varios segundos de densificación el almacenamiento de puntos 3D satura la memoria y se ralentizaba el proceso en exceso. Se planteará una posible solución a este problema en el capítulo donde se tratan las líneas futuras de trabajo.

Tabla 4.2: Tiempos de cómputo: mapa sparse, valores medios por fotograma

	SPARSE		
	Proceso: Tracking		
Secuencia	nº fotogramas	tmedio por frame (s)	tmediana (s)
1	5741	0.0376	-
2	3673	0.0228	-
3	5366	0.0297	-
4	2121	0.0294	0.0298
5	2016	0.0322	0.0305
6	3257	0.0337	0.0339

Tabla 4.3: Tiempos de cómputo: mapa denso, valores medios por KeyFrame. Resolución de las secuencias 720x288 pixels.

	DENSO			
	Mapa de disparidades	Proyección de puntos densos (evitar duplicación)		Obtención coord. 3D globales
Secuencia	tmedio (s)	nºmedio puntos proyectados	tmedio (s)	tmedio (s)
1	0.009950	7385	0.0143	0.00684
2	0.005600	79529	0.0939	0.00590
3	0.005325	19045	0.0535	0.00708
4	0.004990	22368	0.0357	0.00403
5	-	-	-	-
6	0.006050	27666	0.0372	0.00580

4.5.2. Análisis y conclusiones de los tiempos de cómputo

La frecuencia de vídeo es de 20 fotogramas por segundo, por lo que se obtiene un periodo máximo de cómputo de 0.05 segundos antes de la entrada del siguiente fotograma. De este modo, se observa que el proceso de Tracking en el mapa sparse entra dentro del tiempo con un margen suficiente para trabajar a tiempo real.

Por otro lado, en la densificación, los tiempos de la creación del mapa de disparidades y la obtención 3D de las coordenadas de los puntos densos son casi despreciables. Los tiempos más significativos corresponden a la proyección de los puntos ya existentes para evitar duplicaciones. Esto tiene su justificación, no se ha aprovechado para los puntos densos la estructura de datos existente en ORBSLAM diseñada para el almacenamiento de puntos sparse. Se ha evitado su uso puesto que engloba datos e información que se han considerado innecesarios y, por lo tanto, supondría un mayor costo computacional. Al final algunos de esos parámetros y datos han acabado siendo necesarios posteriormente. Este es el caso de los datos respectivos a la orientación de los puntos densos, necesaria para agilizar la proyección del mapa denso en los nuevos

KeyFrames (en concreto el filtro de ORBSLAM denominado “inFrustum”).

Aun con los tiempos de proyección de puntos densos, estos parámetros de tiempo son por cada KeyFrame en vez de por cada fotograma. Entre un KeyFrame y KeyFrame ORBSLAM establece un mínimo 10 fotogramas. Se dispone por lo tanto de 0.5 s para el cómputo gobal de los nuevos puntos densos. Por lo tanto, se trabaja también a tiempo real en el mapa denso (hasta que el mapa de puntos densos es demasiado extenso).

Capítulo 5

Líneas futuras

5.1. Corto plazo

Uno de los problemas a solventar es la ralentización del sistema cuando se acumulan excesivos puntos densos. Sería interesante la mejora de la estructura de datos de almacenamiento de dichos puntos para permitir un filtrado y una proyección más rápidos. Además, se hace evidente la necesidad de separar el proceso de representación de los puntos en un thread diferente para agilizar todo el proceso de cómputo. Se dejaría como proceso de fondo la representación, pero se podría seguir teniendo a tiempo real información sobre las posibles colisiones de la cámara y su orientación con respecto a las superficies. Otra posibilidad sería realizar una optimización utilizando una tarjeta gráfica.

Una cuestión que atañe tanto al sistema SLAM sparse como a la densificación es la comprobación de la calidad de sus resultados. Es necesario, por lo tanto, un ground-trouth con el que comparar tanto la trayectoria de la cámara como la densificación 3D. Existen ya secuencias de modelos 3D de partes del cuerpo humano que van acompañadas de los datos concernientes a la cámara (parámetros y trayectoria) [9]. Estas secuencias pueden hacer de ground-trouth para el sistema SLAM y para la densificación, pudiéndose comparar trayectoria y el modelo 3D obtenidos.

Se ha realizado una pequeña prueba con un modelo propio, de lo que podría ser una escena irregular del interior del cuerpo humano, al que se le han añadido texturas de imágenes reales. También se ha añadido una fuente de luz solidaria a la cámara emulando el tipo de luz que utilizan los endoscopios. La cámara (estéreo) se ha generado con los parámetros más parecidos posible a los que tenían las cámaras de la base de datos del Hamlyn Center. Se muestran, a continuación, imágenes de esta primera aproximación o modelo, comparando también, solo visualmente, la trayectoria de la cámara en el modelo y la obtenida en ORB-SLAM2 stereo:

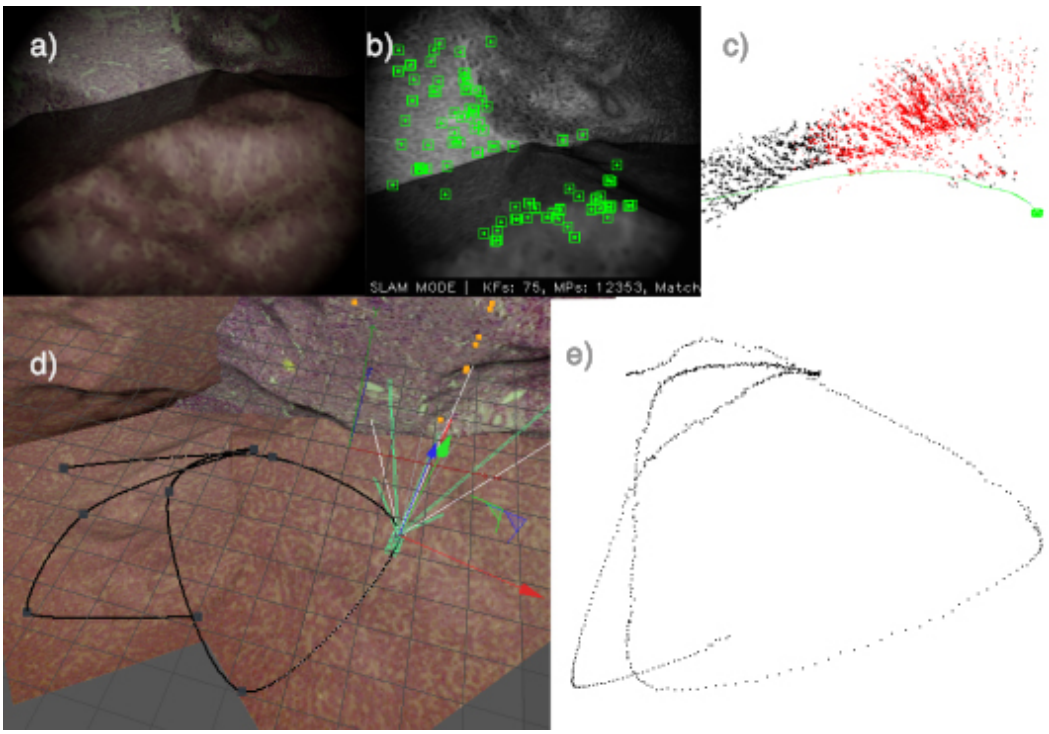

Figura 5.1: Fotograma (cámara izquierda) de la secuencia generada en el modelo (a). Fotograma de la secuencia en ORB-SLAM2 (b). Proceso de generación de la trayectoria en ORBSLAM (c). Vista general del modelo y de la trayectoria creados como ground-truth (d). Puntos de la trayectoria obtenida en ORBSLAM, representados en Meshlab (e).

Se observa que la trayectoria es parecida aunque con ciertas variaciones. Esto no deja de ser una primera aproximación meramente ilustrativa, se procuraría mejorar el modelo y la cámara y se realizaría una comparación cuantitativa entre ambas trayectorias. También se podría realizar una comparación del modelo denso y, yendo un paso más allá, se podrían animar paredes de la escena del modelo. Dicha animación supondría también un ground-truth para el análisis de la calidad de la detección de paredes deformables. Esto permitiría medir errores y realizar las correcciones pertinentes en el programa para conseguir finalmente un modelo denso y una localización con respecto a dicho modelo más fiables, todo ello con medidas cuantificables.

No se plantea en ningún caso validar completamente el sistema con solo pruebas de simulación. Este ground-truth simulado sería el paso o prueba previo a la evaluación del sistema en entornos reales.

5.2. Largo plazo

La meta es la obtención de un modelo denso con un mallado deformable y la orientación con respecto a dicho modelo a tiempo real, lo que permitiría la obtención de

muchos datos, hasta ahora inaccesibles, únicamente con la información visual de las cámaras del endoscopio. Una vez logrado este objetivo se plantearían aplicaciones como:

- Proyección en realidad aumentada del interior del paciente. Realizando previamente un TAC se podría localizar la región de superficie densa generada en ORB-SLAM2 con en el modelo obtenido con el TAC. Posteriormente se proyectaría como realidad aumentada sobre la superficie exterior del paciente (con gafas de realidad aumentada, por ejemplo) pudiendo así el cirujano ver a tiempo real todo lo que sucede y facilitar también el uso automatizado de herramientas quirúrgicas.
- Eliminación del movimiento relativo entre superficies y la cámara para facilitar las intervenciones. Se podría simular, por ejemplo, una operación a "corazón parado" aún estando este latiendo si se aplicase a la cámara, y por lo tanto también a las herramientas, el mismo movimiento relativo que tiene la superficie a tiempo real.

Capítulo 6

Bibliografía

[1] Raul Mur-Artal and Juan D Tardós. ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 2017.

[2] Iñigo Cirauqui. Evaluación de orbslam en secuencias de endoscopia médica. *Trabajo Fin de Grado, Ingeniería de Tecnologías Industriales*, 2016.

[3] 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.

[4] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2, 2010.

[5] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[6] Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly, 2016.

[7] Itseez. Open source computer vision library. `https://github.com/itseez/opencv`, 2015. [Online; accedido el 20 de mayo de 2017].

[8] Peter Mountney, Danail Stoyanov, and Guang-Zhong Yang. Three-dimensional tissue deformation recovery and tracking. *IEEE Signal Processing Magazine*, 27(4):14–24, 2010.

[9] Sebastian Röhl, Sebastian Bodenstedt, Stefan Suwelack, Hannes Kenngott, Beat P Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Dense gpu-enhanced surface reconstruction from stereo endoscopic images for intraoperative registration. *Medical physics*, 39(3):1632–1645, 2012.

[10] Itseez. *The OpenCV Reference Manual*, 2.4.9.0 edition, April 2014.

Anexos

Anexos A

Ejemplo de un “hamlyn.yaml”.

Ejemplo de archivo generado tras la ejecución de “StereoVidsToFrames”.

```
%YAML:1.0
#--------------------------------------------------------------------------------
# Stereo Rectification. Only if you need to pre-rectify the images.
#--------------------------------------------------------------------------------
LEFT.K: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data:[839.042847, 0.000000, 334.948578, 0.000000, 461.469147, 131.451157, 0.000000, 0.000000, 1.000000]
# Camera: left distortion coef
LEFT.D: !!opencv-matrix
   rows: 1
   cols: 4
   dt: d
   data:[-0.144240, 0.031213, -0.009081, -0.008255]
# Camera: right intrinsics
RIGHT.K: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data:[831.130798, 0.000000, 401.026520, 0.000000, 449.551483, 117.013893, 0.000000, 0.000000, 1.000000]
# Camera: right distortion coef
RIGHT.D: !!opencv-matrix
   rows: 1
   cols: 4
   dt: d
   data:[-0.267182, 0.524761, 0.000382, -0.009791]
# R matrix
R.matrix: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data:[0.999850, 0.003814, 0.016909, -0.003944, 0.999963, 0.007654, -0.016880, -0.007720, 0.999828]
# T matrix
T.matrix: !!opencv-matrix
   rows: 3
   cols: 1
   dt: d
   data:[-5.596993, -0.213306, -1.175782]
RIGHT.height: 288
RIGHT.width: 720
LEFT.height: 288
LEFT.width: 720
MAT.Q:  !!opencv-matrix
   rows: 4
   cols: 4
   dt: d
   data:[1, 0, 0, -263.9191370010375977, 0, 1, 0, -128.1469240188598633, 0, 0, 0, 360.2022656779874978, 0,
0, 0.1747293585229918178, -0]
LEFT.R:  !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data:[0.9741972619910596043, 0.03941345583702905719, 0.2222302279801410818, -0.04027796686546910448,
0.9991883068774588539, -0.0006424831058389058984, -0.2220751677119869039, -0.008325076476505996817,
0.9749940066417603912]
RIGHT.R:  !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data:[0.9779589965476757074, 0.03727082054910538927, 0.2054436346228810384, -0.03647181507727277616,
0.9993052014985729937, -0.007675999144454570891, -0.2055869834800654328, 1.391017004309267821e-05,
0.9786388465772623002]
```

```
LEFT.P:  !!opencv-matrix
   rows: 3
   cols: 4
   dt: d
   data:[360.2022656779874978, 0, 263.9191370010375977, 0, 0, 360.2022656779874978, 128.1469240188598633, 0,
0, 0, 1, 0]
RIGHT.P:  !!opencv-matrix
   rows: 3
   cols: 4
   dt: d
   data:[360.2022656779874978, 0, 263.9191370010375977, -2061.486797197794203, 0, 360.2022656779874978,
128.1469240188598633, 0, 0, 0, 1, 0]
#--------------------------------------------------------------------------------------------
# Camera parameters
# Camera calibration and distortion parameters (OpenCV)
Camera.fx: 360.2022657
Camera.fy: 360.2022657
Camera.cx: 263.919137
Camera.cy: 128.146924
Camera.k1: 0.0
Camera.k2: 0.0
Camera.p1: 0.0
Camera.p2: 0.0
Camera.width: 720
Camera.height: 288
# Camera frames per second
Camera.fps: 20
# stereo baseline times fx
Camera.bf: 2061.486797197794203
# Color order of the images (0: BGR, 1: RGB. It is ignored if images are grayscale)
Camera.RGB: 1
# Close/Far threshold. Baseline times.
ThDepth: 35
#--------------------------------------------------------------------------------------------
# ORB Parameters
#--------------------------------------------------------------------------------------------
# ORB Extractor: Number of features per image
ORBextractor.nFeatures: 1200
# ORB Extractor: Scale factor between levels in the scale pyramid
ORBextractor.scaleFactor: 1.2
# ORB Extractor: Number of levels in the scale pyramid
ORBextractor.nLevels: 8
# ORB Extractor: Fast threshold
ORBextractor.iniThFAST: 20
ORBextractor.minThFAST: 7
#--------------------------------------------------------------------------------------------
# Viewer Parameters
#--------------------------------------------------------------------------------------------
Viewer.KeyFrameSize: 0.2
Viewer.KeyFrameLineWidth: 1
Viewer.GraphLineWidth: 0.9
Viewer.PointSize: 2
Viewer.CameraSize: 5.723136675216573543
Viewer.CameraLineWidth: 3
Viewer.ViewpointX: 0
Viewer.ViewpointY: -0.7
Viewer.ViewpointZ: -1.8
Viewer.ViewpointF: 500
```

Anexos B

Repositorio Github

Todo el código desarrollado se encuentra en un repositorio institucional privado de Github de la Universidad de Zaragoza para facilitar su consulta y posibilitar su modificación en un futuro.

`https://github.com/UZ-SLAMLab/ORB_SLAM2_Stereo_Densify_Endoscopy_Hamlyn_Dataset`

Lista de Figuras

2.1. Trayectoria monocular (en verde). Se muestra la localización en los fotogramas 1 y 2 del punto 3D denominado X correspondiente al pico de la pirámide proyectado con unas coordenadas (en píxeles) u y v. En gris, morado y cian los haces de luz y sus intersecciones en algunos puntos de interés. 10

2.2. Obtención del la profundidad del estéreo: Donde la Baseline V, las distancias focales f y las coordenadas del punto en las proyecciones (x y x') son datos conocidos y $Z = \frac{Bf}{X-X'}$. 12

2.3. Trayectoria estéreo (en verde): La profundidad del punto 3D X se obtiene tanto en el fotograma uno como en el dos mediante la triangulación del par estéreo correspondiente a cada fotograma. En gris, morado y cian los haces de luz y sus intersecciones en algunos puntos de interés. . 13

2.4. Vista en planta de una escena. Verde: la trayectoria de la cámara y los fotogramas normales (pares estéreo). Azul oscuro: de mayor tamaño los KeyFrames (pares estéreo también). Se muestran también dos cortes coplanarios de los campos de visión de los dos primeros keyframes (correspondientes a sus cámaras izquierdas). 16

2.5. Representación del proceso de Loop Closing 17

3.1. Par sin rectificar: plano epipolar y líneas epipolares para un punto X 3D y posibles posiciónes de su correspondencia en la cámara derecha x'. . . 19

3.2. Proceso de eliminación de distorsión y de rectificación: (a)Imágenes en bruto, (b)Eliminación de la distorsión y (c)Rectificación (ilustración del libro Bradski [6]). 20

3.3. Par estéreo de una secuencia de la base de datos Hamlyn antes y después de la rectificación y la eliminación de la distorsión. Se puede apreciar cómo tras la rectificación, se encuentran correspondencias siempre en la misma horizontal (C1 y C2 cámaras izquierda y derecha respectivamente). 21

3.4. Comparación de mapa denso con diferentes valores del parámetro blockSize. Se observan demasiados puntos dispersos para el valor de 5 píxeles (valor mínimo) y demasiados huecos con valores demasiado altos como 45 píxeles. El valor de 17 consigue bastante densidad y suficiente detalle. 21
3.5. Mapa denso local tras el recorte del marco. Los puntos negros y rojos superpuestos corresponden a puntos del mapa sparse. 22

4.1. Esquema de programas (azul) y entradas/salidas (amarillo). 25
4.2. Secuencia 1. 29
4.3. Secuencia 2. 30
4.4. Secuencia 3. 31
4.5. Secuencia 4. 32
4.6. Secuencia 5. 33
4.7. Secuencia 6. 34
4.8. Comparaciones de los mapas sparse y densos de las secuencias 3,4,5 y 6. 35

5.1. Fotograma (cámara izquierda) de la secuencia generada en el modelo (a). Fotograma de la secuencia en ORB-SLAM2 (b). Proceso de generación de la trayectoria en ORBSLAM (c). Vista general del modelo y de la trayectoria creados como ground-truth (d). Puntos de la trayectoria obtenida en ORBSLAM, representados en Meshlab (e). 40

Lista de Tablas

2.1. Monocular: grados de libertad del sistema en función del número de puntos (correspondencias). 11
2.2. Estéreo: grados de libertad del sistema estéreo en función del número de puntos obtenidos (correspondencias estéreo). 14

4.1. Tabulación de los parámetros modificados para la sintonización del programa. 26
4.2. Tiempos de cómputo: mapa sparse, valores medios por fotograma . . . 37
4.3. Tiempos de cómputo: mapa denso, valores medios por KeyFrame. Resolución de las secuencias 720x288 pixels. 37