Trabajo Fin de Máster en Ingeniería Informática

Navegación endoscópica con ORB-SLAM para cirugía uretral mínimamente invasiva

Endoscopic navigation using ORB-SLAM for urethral minimally-invasive surgery

Autor

LAURA OLIVA MAZA

Director

JOSE MARÍA MARTÍNEZ MONTIEL

Codirector

KLAUS STROBL

Escuela de Ingeniería y Arquitectura

2019-2020

Resumen

En las operaciones de cirugía mínimamente invasiva un endoscopio se mueve por una cavidad u órgano del cuerpo. El objetivo del SLAM (*Simultaneous Localization And Mapping*) es estimar el mapa 3D de la cavidad u órgano que se está explorando y localizar el endoscopio con respecto al mapa. Para ello la única información empleada es el vídeo tomado por el endoscopio.

ORB-SLAM es un sistema de SLAM desarrollado en la Universidad de Zaragoza que utiliza puntos FAST y descriptores ORB para estimar tanto el mapa 3D como la localización de la cámara. El principal problema de ORB-SLAM2 es que este sistema sólo funciona de forma robusta (con una precisión alta y sin fallos o desviaciones) en escenas que se pueden asumir rígidas, con textura, en las que es fácil extraer y emparejar puntos característicos visuales. Éste no es el caso de los órganos, pues son deformables, húmedos y es difícil encontrar visualmente puntos característicos fiables (esquinas) dentro de éstos, lo que puede llevar a una estimación incorrecta del mapa. A ello hay que añadir los cambios bruscos de iluminación y los movimientos bruscos del endoscopio.

En este trabajo se han realizado diferentes modificaciones a ORB-SLAM2 para que funcione en secuencias grabadas por un endoscopio cómo la reducción del número de puntos necesarios para realizar el seguimiento, la creación de un mapa inicial más robusto y el pre-procesamiento de las imágenes para aumentar el contraste, reducir el ruido y evitar detectar puntos en los reflejos. Además, para conseguir puntos más robustos se han probado diferentes combinaciones de detector-descriptor creando un nuevo vocabulario para cada una de ellas. Se ha demostrado que con las modificaciones realizadas a ORB-SLAM2, el cambio de FAST-ORB a A-KAZE-ORB y el nuevo vocabulario se ha conseguido estimar un 80 % de la trayectoria cuando con el sistema original no se llegaba a estimar un 40 %. Esta mejora sólo ha supuesto un incremento en el tiempo de cómputo de 40 ms.

Un sistema de SLAM puede ser muy útil a la hora de crear una interfaz quirúrgica ya que éste permite conectar un punto que se toca en la pantalla con el mapa 3D estimado permitiendo al cirujano insertar anotaciones de realidad aumentada en los lugares que se crean oportunos.

Abstract

In minimally invasive surgery, an endoscope is moved through a body cavity or organ. The aim of SLAM (*Simultaneous Localization And Mapping*) is to estimate the 3D map of the cavity or organ being explored and to locate the endoscope with respect to the map. The only information used for this purpose is the video taken by the endoscope.

ORB-SLAM is a SLAM system developed at the University of Zaragoza that uses FAST features and ORB descriptors to estimate both the 3D map and the camera pose. The main problem of ORB-SLAM2 is that this system is robust (with high accuracy and without failures or deviations) when it works with scenes that can be assumed rigid, with texture, in which it is easy to extract and match visual features. This is not the case with organs, as they are deformable, wet and it is difficult to visually find reliable features (corners) within them, which can lead to an incorrect map estimation. Abrupt changes in illumination and sudden movements of the endoscope must also be taken into account.

In this work, ORB-SLAM2 has been modified to work in sequences recorded by an endoscope. This modifications include reducing the number of points required to perform the monitoring, creating a more robust initial map and pre-processing the images to increase contrast, reduce noise and avoid detecting points in the reflections. In addition, to achieve more robust keypoints, different detector-descriptor combinations have been tested, creating a new vocabulary for each of them. It has been demonstrated that with the modifications made to ORB-SLAM2, the change from FAST-ORB to A-KAZE-ORB and the new vocabulary, it has been possible to estimate 80 % of the trajectory when with the original system it was not possible to estimate 40 %. This improvement has only increased the computation time by 40 ms.

A SLAM system can be very useful in surgical interfaces since it allows to connect a point in screen with the estimated 3D map, allowing the surgeon to insert augmented reality annotations in the appropriate places.

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe entregarse en la Secretaría de la EINA, dentro del plazo de depósito del TFG/TFM para su evaluación).

D./Dª. Laura Oliva Maza ,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) Máster

(Título del Trabajo)

Navegación endoscópica con ORB-SLAM para cirugía uretral mínimamente invasiva

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

Zaragoza, 22/06/2020

Fdo: Laura Oliva Maza

Agradecimientos

Lo primero de todo me gustaría dar las gracias a Jose María Martínez Montiel por haberme dado la oportunidad de realizar este trabajo y por su ayuda.

También quería agradecer a Klaus, Álex e Irene por su ayuda y por haber hecho que en Múnich me sintiera como si estuviera en casa.

Por último, agradecer a mi familia y amigos que, a pesar de la distancia, siempre han estado allí para alegrarme el día acortando las distancias con una simple videollamada.

El Trabajo de Fin de Máster se ha llevado a cabo con el apoyo financiero del Ministerio de Economía y Competitividad del Estado Español mediante los proyectos "DPI2017-91104-EXP:SLAM Visual Deformable para Endoscopia" y "PGC2018-096367-B-I00: SLAM Visual Mejorado mediante aprendizaje profundo"

Índice general

Capítulo 1

Introducción y objetivos

1.1. Introducción

El campo de la medicina está en constante desarrollo. Aproximadamente hace 150 años se realizó la primera operación con éxito y hoy en día es un proceso muy habitual que se realiza varias veces al día en los diferentes hospitales del mundo. Desde esta primera operación la cirugía ha ido evolucionando para facilitar este proceso al cirujano y para causar menos heridas al paciente y facilitar su recuperación. Para la segunda parte, a mediados de los años 80 nace la cirugía mínimamente invasiva. El objetivo de ésta es disminuir el número de incisiones realizadas al paciente a la hora de realizar una operación. En este tipo de cirugía se realizan una o más pequeñas incisiones en el cuerpo y se introduce una pequeña cámara (laparoscopio o endoscopio) a través de una de estas incisiones o una abertura natural a fin de guiar la cirugía. Hoy en día este tipo de cirugía enmarca casi todas las disciplinas quirúrgicas. Además, las innovaciones tecnológicas y robóticas permiten que este tipo de cirugía evolucione más rápido, remplazando progresivamente a la cirugía convencional. En la figura 1.1 se puede observar que la cirugía mínimamente invasiva (laparoscopia es la más común) ha ido sustituyendo a la cirugía convencional o cirugía abierta a lo largo de los años.

En la cirugía mínimamente invasiva se utiliza un endoscopio para guiar al cirujano a lo largo de la operación. Un endoscopio (figura 1.2) es un instrumento que se utiliza para la exploración visual de los conductos o cavidades del cuerpo humano. Consiste en un tubo provisto con una cámara y un sistema de iluminación en el extremo de éste. Este tubo también tiene un pequeño canal por el que se pueden introducir diferentes herramientas. Este tubo puede ser flexible o rígido. En la navegación endoscópica o en una intervención endoscópica, el cirujano se guía por la información visual proporcionada por el endoscopio y por los datos obtenidos en la fase preoperatoria. Sin embargo, la cámara tiene que ser guiada por el cirujano o sus ayudantes, aumentando la de ya de por sí carga en el equipo médico.

Las principales ventajas de la cirugía mínimamente invasiva son:

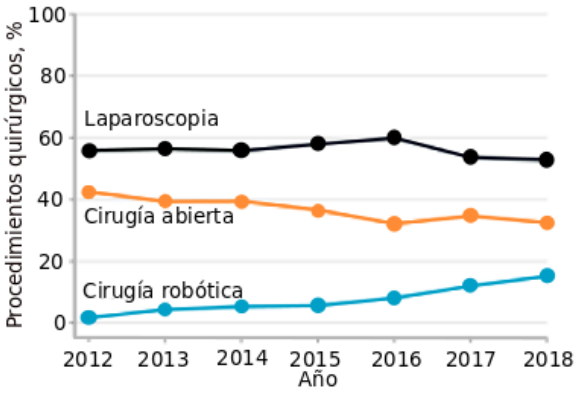


Figura 1.1: Porcentaje de operaciones en cada tipo de cirugía[1].

Figura 1.2: Endoscopio flexible usado en uretereoscopia.

- La disminución del dolor postoperatorio debido a la ausencia de incisiones quirúrgicas importantes y a la reducción del trauma en los tejidos sanos.
- El menor número de complicaciones en la herida quirúrgica.
- La disminución del postoperatorio y estancia en el hospital. En consecuencia, se reducen los costes asociados y las listas de espera.
- Los resultados estéticos altamente mejorados.
- La rehabilitación acelerada del paciente con una vuelta a la normalidad más rápida.

Las desventajas de este tipo de cirugía están ligadas a las dificultades técnicas que conlleva. Debido a que en este tipo de operaciones el cirujano trabaja a partir de las imágenes 2D del endoscopio

[1]Imagen obtenida de `https://mimicsimulation.com/the-growth-of-robotic-surgery/`

visualizadas en los monitores (figura 1.3), hay una pérdida del área de visión 2D (campo de visión), de la percepción espacial (visión 3D) y de la percepción táctil (no existe palpación de los tejidos). Además en este tipo de operaciones el cirujano encuentra dificultades a la hora de determinar si la zona del órgano que está explorando es una zona nueva o es una zona que ya ha explorado debido a la reducida área de visión y al oculto efecto palanca del endoscopio. Otra de las dificultades que el cirujano puede encontrar en este tipo de operaciones es que el endoscopio es extraído y re-introducido varias veces a lo largo de la operación por lo que cuando el endoscopio es re-introducido el cirujano tiene que ser capaz de volver al punto donde estaba operando antes de extraer el endoscopio.

Figura 1.3: Sala de operaciones.

Para ayudar al cirujano en este tipo de operaciones existen diferentes técnicas de visión por computador que se pueden aplicar en el campo de la medicina. Usando técnicas de visión por computador se puede estimar el mapa 3D del órgano que se esta explorando, estimar la posición y orientación del endoscopio o añadir diferentes objetos virtuales ya sea con información de la fase preoperatoria o con información obtenida a lo largo de la operación. Además, también se pueden usar diferentes técnicas para identificar en tiempo real estructuras que no siempre son fáciles de identificar o que están situadas en planos profundos. Estas técnicas de visión por computador para ayudan a paliar las desventajas de la cirugía mínimamente invasiva y a realizar las operaciones de forma más sencilla y precisa.

1.2. Objetivos y alcance del proyecto

El objetivo principal de este trabajo es desarrollar un sistema capaz de estimar un mapa de puntos 3D de la cavidad observada y localizar el endoscopio con respecto a este mapa utilizando exclusivamente las imágenes tomadas por endoscopio (SLAM visual, VSLAM, *Visual Simultaneous Localization and Mapping*)).

Durante las operaciones de cirugía mínimamente invasiva, el endoscopio es extraído y re-introducido varias veces. Cuando el endoscopio es extraído, la localización de éste se pierde y cuando es re-introducido, el sistema debe ser capaz de estimar la localización de éste con respecto al mapa utilizando

únicamente la información visual. Este proceso es conocido como relocalización. La relocalización es importante en este tipo de operaciones ya que el cirujano encuentra dificultades a la hora de volver al punto dónde estaba operando antes de extraer el endoscopio. Al re-introducir el endoscopio, éste puede tener una orientación diferentes a la anterior y para el ojo humano es difícil diferenciar entre las partes de un órgano con el reducido campo de visión del endoscopio. Un sistema de VSLAM podría ayudar al cirujano con este tipo de dificultades.

Para estimar el mapa 3D y la posición y orientación del endoscopio en cada instante se va a partir de un sistema de VSLAM llamado ORB-SLAM2. ORB-SLAM2 es una solución propuesta al problema de SLAM visual desarrollada en la Universidad de Zaragoza. ORB-SLAM2 utiliza los puntos naturales ORB extraídos de las imágenes capturadas por la cámara para estimar tanto el mapa 3D como la trayectoria de la cámara. ORB-SLAM2 es un sistema de SLAM que hoy en día se utiliza en una gran variedad de proyectos en el mundo entero debido a su precisión, a su funcionamiento en tiempo real y a su licencia GPLV3.

El principal problema de ORB-SLAM2 es que este sistema sólo funciona de forma robusta (con una precisión alta y sin fallos o desviaciones) en escenas que se pueden asumir rígidas, con textura, en las que es fácil extraer y emparejar puntos característicos visuales. Éste no es el caso de los órganos, pues son deformables (respiración, latidos, movimientos del paciente, etc.) y es difícil encontrar visualmente puntos característicos fiables (esquinas) dentro de éstos, lo que puede llevar a una estimación incorrecta del mapa. A ello hay que añadir los cambios bruscos de iluminación y los movimientos bruscos del endoscopio.

En este trabajo se va a modificar ORB-SLAM2 para que sea capaz de estimar de forma robusta la posición y orientación ("*pose*") del endoscopio y el mapa 3D de puntos naturales del órgano que se está examinando.

Los objetivos de este trabajo son:

1. El estudio detallado del algoritmo ORB-SLAM2 y sus limitaciones en cirugía.

2. La optimización del algoritmo ORB-SLAM2 en cuanto a la elección y uso de los diferentes detectores y extractores de puntos de interés existentes en la literatura.

3. La modificación de ORB-SLAM2 para conseguir un funcionamiento correcto en cirugía mínimamente invasiva. En concreto, en el procedimiento de eliminación de pólipos de la vejiga.

4. El análisis y la evaluación de las prestaciones de la nueva versión de ORB-SLAM2 en secuencias médicas representativas.

1.3. Herramientas utilizadas

Este trabajo parte de código de ORB-SLAM2 presente en su repositorio de GitHub (`https://github.com/raulmur/ORB_SLAM2`, último *commit* `f2e6f51` el 11 Oct 2017). En esta librería se utilizan las siguientes herramientas:

- C++: lenguaje de programación de ORB-SLAM2.
- DBoW: librería que usa ORB-SLAM2 para la creación de la bolsa de palabras necesaria para comparar imágenes.
- Pangolin: librería que usa ORB-SLAM2 para la visualización.
- OpenCV: librería de código abierto en la que se pueden encontrar diferentes algoritmos de visión por computador.
- Git + GitHub: herramienta utilizada para el control de versiones de software.
- Bash: lenguaje de comandos usado para la automatización de experimentos.
- Matlab: software utilizado para la creación de las gráficas presentes en este documento.
- LaTeX: sistema de composición de textos utilizado para la redacción de este documento.
- Google Scholar: motor de búsqueda utilizado para la obtención de documentos.

1.4. Estructura del documento

Este documento se ha estructurado de la siguiente forma:

- Capítulo 2: Estado del arte. En este capítulo se explicarán las técnicas de visión por computador actuales utilizadas en la cirugía mínimamente invasiva y en la estimación del movimiento de la cámara en general.
- Capítulo 3: Método de seguimiento de la posición y orientación (*pose*). En este apartado se explicará con más detalle el software ORB-SLAM2 detallando los problemas encontrados al usar este software en órganos y las modificaciones realizadas.
- Capítulo 4: Evaluación. En este apartado se explicarán los experimentos realizados para la evaluación del nuevo sistema y los resultados obtenidos. Además, se presentará la visión de una interfaz para el cirujano utilizando los datos obtenidos.
- Capítulo 5: Conclusiones y trabajo futuro. En este apartado se explicarán las conclusiones finales de este trabajo y las posibles líneas futuras de investigación en base a los resultados.

Capítulo 2

Estado del arte

En este capítulo se va a hacer una pequeña introducción de los diferentes campos que abarca este trabajo y se referenciará a las técnicas más relevantes para este trabajo.

2.1. Retroalimentación visual para cirugía mínimamente invasiva

La endoscopia es un proceso en el que el cirujano utiliza material especializado para ver y operar dentro del cuerpo humano. Este tipo de operaciones permite al cirujano operar sin la necesidad de realizar grandes incisiones a lo largo del cuerpo ya que para este tipo de operaciones se realiza una o varias pequeñas incisiones para introducir el endoscopio. El endoscopio también puede ser introducido por los orificios naturales del cuerpo como por ejemplo la boca, el ano, la uretra o la vagina. La endoscopia es un tipo de cirugía mínimamente invasiva, en la que se realizan pequeñas incisiones por las que se introduce un tubo con una cámara en el extremo llamado endoscopio. De esta forma el cirujano se guía por las imágenes obtenidas por el endoscopio visualizadas en un monitor. En este tipo de cirugía, el endoscopio es extraído y re-introducido varias veces a lo largo de la operación. Para no dañar los conductos y los órganos que conectan el órgano en el que se va a realizar la operación con el exterior del cuerpo, primero se introduce un tubo vacío por el que irá el endoscopio.

La endoscopia es un operación que se ha extendido a la mayoría de los campos dentro de la cirugía debido a las diferentes ventajas comentadas en el capítulo anterior. Por este motivo existen diferentes tipos de endoscopia dependiendo de la zona en la que se vaya a operar como por ejemplo:

- Citoscopia. Durante la citoscopia el urólogo inserta el citoscopio (tubo delgado con una cámara y una luz en el extremo, se puede usar endoscopio flexible o rígido) a través de la uretra para examinar la vejiga y eliminar tumores, pólipos o cualquier tipo de tejido anormal.

- Ureteroscopia. La ureteroscopia es un procedimiento para tratar las piedras en el riñón. En este

procedimiento el urólogo introduce un uteroscopio (endoscopio flexible) a través de la uretra y la vejiga, subiendo por el uréter hasta llegar al riñón.

- Laparoscopia. La laparoscopia es un procedimiento quirúrgico de diagnóstico usado para examinar los órganos dentro del abdomen. En este procedimiento se examina la parte exterior de los órganos. Para ello se realizan pequeñas incisiones (típicamente tres) cerca de la zona a examinar y se introduce el laparoscopio por una ellas. En este tipo de operaciones se utiliza un endoscopio rígido llamado laparoscopio que puede llegar a ser de mayor tamaño y tener mejor calidad de imagen.

En este trabajo nos vamos a centrar en la citoscopia (extración de pólipos en la vejiga) y la ureteroscopia (eliminación de piedras del riñón). Ambas operaciones siguen un procedimiento similar. Primero hay una fase en la que se examina el órgano, localizando las piedras o los pólipos que hay que extraer. Luego el endoscopio es extraído para introducir por el canal del endoscopio la herramienta necesaria y se vuelve a introducir para empezar el procedimiento. Durante este procedimiento el endoscopio es extraído y re-introducido varias veces para completar la extracción de las piedras en el riñón o de los pólipos en la vejiga. En la operación de eliminación de piedras en el riñón, éstas se han de romper, creando polvo que nubla la visión de la cámara. Mientras que en la operación de eliminación de pólipos o tejidos anormales de la vejiga, éstos se queman para separarlos de la superficie de la vejiga y extraerlos, por lo que las imágenes de éste tipo de operaciones son más limpias. Además, en la citoscopia se puede utilizar un endoscopio rígido cuya calidad de imagen es más alta que la de un endoscopio flexible.

En ambos procedimientos se podría usar un sistema de SLAM visual para crear una interfaz que ayude al cirujano a realizar las operaciones o a manejar el brazo robótico de forma más precisa. Este sistema debería ser capaz de estimar la localización del endoscopio y el mapa de puntos 3D del órgano que se está examinando de forma robusta. Además, debería ser capaz de realizar relocalización para que al volver a introducir el endoscopio el sistema sepa dónde está el endoscopio con respecto al mapa.

En este proyecto se va a trabajar con secuencias obtenidas por un endoscopio flexible. Estas secuencias han sido grabadas en una citoscopia y en una ureteroscopia. En el apartado 4.1 se entrará en más detalle en los diferentes *datasets* utilizados.

2.2. Localización visual

La localización visual es el problema de estimar la posición y orientación desde la que se tomó una imagen, es decir, la posición y orientación de la cámara correspondiente con respecto a alguna representación de la escena o una situación anterior de la cámara. La localización visual es una tecnología clave para aplicaciones como la realidad aumentada o la robótica.

La localización por odometría visual no requiere de mapa global explícito. La localización y modelado simultáneos (SLAM, *Simultaneous Localization And Mapping*) es una técnica usada para construir el mapa 3D del entorno desconocido que rodea a una cámara a la vez que se estima su trayectoria. Este mapa promete mejorar la precisión respecto a la odometría visual. SLAM visual (*vSLAM*, visual SLAM) consiste en el empleo de una o varias cámaras para desarrollar el mapeado y localización simultáneos utilizando únicamente la información obtenida de las imágenes. Este tipo de algoritmos están compuestos fundamentalmente por cinco módulos diferentes: inicialización, localización, mapeado, re-localización y por último la optimización del mapeado. Algunos de los algoritmos de SLAM Monocular más conocidos son PTAM, LSD-SLAM y ORB-SLAM. Estos modelos están basados en *keyframes*, es decir, no se trabaja con todas las imágenes capturadas por la cámara sino sólo con aquellas que se considera que tienen suficiente información nueva (*keyframes*). De esta forma se reduce el tiempo de cómputo.

- PTAM [1] fue el primer algoritmo en implementar la idea de *keyframes*. PTAM utiliza puntos FAST y los empareja utilizando un parche de correlación.
- LSD-SLAM [2] es capaz de construir mapas semi-densos a gran escala usando métodos directos en los que se trabaja con la intensidad de los píxeles de la imagen en lugar de realizar un seguimiento de los puntos de interés visuales detectados en la imagen.
- ORB-SLAM [3] utiliza puntos FAST y descriptores ORB para realizar el seguimiento de los puntos. Los descriptores ORB son descriptores binarios invariantes a la rotación y a la escala y serán explorados en la sección 3.2.1. El tiempo de cómputo y emparejamiento de estos descriptores es pequeño. ORB-SLAM funciona con cámaras monoculares (una sola cámara), ORB-SLAM2 [4] es la actualización de ORB-SLAM en la que añade el uso de cámaras estéreo (dos cámaras) y cámaras RGB-D (cámaras que también proporcionan información de la profundidad de la imagen.

Una característica importante de estos sistemas es la relocalización y los cierres de bucle. Para conseguir realizar la relocalización o para detectar cierres de bucle se utiliza la técnica de *place recognition* o reconocimiento de lugar. Esta técnica se utiliza para reconocer un lugar, o emparejar dos imágenes, usando únicamente la información visual, sin tener en cuenta la trayectoria de la cámara, la posición y orientación de esta o el mapa 3D construido.

Un ejemplo sería el sistema FAB-MAP [5]. Este sistema detecta bucles con una cámara omnidireccional con un *recall* del 48.4 % y sin falsos positivos. Este sistema representa las imágenes con bolsas de palabras y utiliza el árbol Chow-Liu para aprender de forma *offline* la probabilidad de covisibilidad de las palabras. El problema de este sistema es que la robustez disminuye cuando las imágenes muestran

estructuras muy similares durante un largo periodo de tiempo, como puede ser el caso en el que se utilizan cámaras frontales.

DBoW2 [6] fue el primer sistema que creó la bolsa de palabras a partir de descriptores binarios. Esto hace que disminuya el tiempo de cómputo. ORB-SLAM2 utiliza este sistema con descriptores ORB para realizar la relocalización y la detección de cierres de bucle. Además, ORB-SLAM2 también tiene un grafo de covisibilidad en el que conecta los *keyframes* que tienen información del mismo sitio. ORB-SLAM2 combina la bolsa de palabras con el grafo de covisibilidad para obtener el *keyframe* que más se parece a la imagen actual permitiendo realizar los cierres de bucle y la relocalización.

La relocalización es un proceso que se realiza en cada imagen nueva cuando el sistema está perdido. Para ello utiliza la bolsa de palabras para ver cual es el *keyframe* almacenado en la base de datos más similar a la imagen actual. Si encuentra un *keyframe* lo suficientemente similar, utiliza RANSAC para comprobar si es soportado por suficientes *inliers*. El cierre de bucle es un proceso que, a diferencia de la relocalización, se realiza en cada *keyframe* nuevo. Además, este proceso sólo se realiza en el caso en el que la bolsa de palabras haya detectado tres candidatos consecutivos consistentes. De esta forma se asegura que no haya falsos positivos y que no se cierren bucles cuando no corresponde. Se entrará en más detalle en el proceso de relocalización en la sección 3.4.3 y en el proceso de cierre de bucle en la sección 3.4.4.

En este proyecto se ha trabajado con ORB-SLAM2 para solucionar el problema de localización visual en secuencias médicas. En consecuencia, se ha usado DBoW2 para crear los nuevos vocabularios. En la sección 3.4 se entrará en más detalle sobre estos procedimientos.

2.3. Seguimiento de puntos naturales de interés y relocalización

El seguimiento consiste en detectar los puntos característicos visuales en una imagen y "perseguirlos" a lo largo de varias imágenes o de un vídeo. El principal problema es emparejar estos puntos entre dos imágenes.

Para realizar el seguimiento de puntos característicos, lo primero que se ha de hacer es detectar éstos. Una vez obtenidas los puntos característicos, éstos son descritas por un descriptor. Estos descriptores son los que se utilizan para realizar los emparejamientos. Los detectores y descriptores más comunes son los siguientes:

- Shi-Tomasi [7]: basado en el detector de esquinas de Harris. Este detector utiliza los valores propios del para determinar si un punto es una esquina.
- SIFT (Scale Invariant Feature Transform) [8]: el detector está basado en la Diferencia de Gaussianas (DoG) que es una aproximación de la Laplaciana de una Gaussiana (LoG). Los puntos

característicos se detectan buscando el máximo local utilizando DoG en varias escalas de las imágenes (representación piramidal de imágenes, sección 3.2.3). El descriptor es obtenido extrayendo un vecindario de 16x16 píxeles alrededor de cada punto característico y segmentando la región en sub-bloques. SIFT es, en teoría, invariante a las rotaciones, a la escala y a cambios de afín limitados. Su principal inconveniente es el alto coste de computación.

- SURF (Speeded Up Robust Features) [9]: el detector se basa en el determinante de la matriz Hessiana, usando imágenes integrales para mejorar la velocidad de detección. El descriptor describe cada punto característico con una distribución de respuestas de ondas Haar dentro de cierto vecindario. SURF es invariante a la rotación y a la escala. Sin embargo, tiene poca invarianza ante los cambios de afín. La principal ventaja de SURF sobre SIFT es que es más rápido de calcular.

- A-KAZE (Accelerated KAZE) [10]: A-KAZE explota el espacio de escala no lineal. Este espacio se construye utilizando Fast Explicit Diffusion (FED). El uso de este espacio hace que el desenfoque en las imágenes se adapte localmente a los puntos característicos, reduciendo así el ruido y conservando los detalles. El detector A-KAZE se basa en el determinante de la matriz Hessiana. El descriptor A-KAZE se basa en el algoritmo Binario de la Diferencia Local Modificada (Modified Local Difference Binary, MLDB). A-KAZE es invariante a la escala, a la rotación, a afines limitados y tiene mayor distintividad en escalas variables debido a los espacios de escala no lineal. Se entrará en más detalle en la sección 3.2.2.

- ORB (Oriented FAST and Rotated Brief) [11]: este algoritmo es una mezcla del método de detección FAST (Features from Accelerated Segment Test) [12] modificado y BRIEF (Binary Robust Independent Elementary Features) [13] modificado. Las esquinas FAST se detectan en cada capa de la pirámide de escala (para hacerlo invariante a la escala) y la "esquinidad" de los puntos detectados se evalúa utilizando la puntuación del método de la detección de esquinas de Harris para filtrar los puntos de mayor calidad. Como el descriptor BRIEF no es invariante a los cambios de rotación, ORB utiliza una modificación de éste en el que se calcula la orientación principal de cada punto para hacerlo invariante a los cambios de rotación. ORB es invariable a la escala, rotación y a cambios afines.

Una vez obtenidos los puntos característicos y sus descriptores, se han de emparejar los puntos característicos de diferentes imágenes. Para ello se comparan los descriptores y se decide si dos descriptores son iguales o no. Uno de los algoritmos básicos sería por fuerza bruta en el que se compara cada descriptor de una imagen con cada descriptor de la otra. Otro algoritmo sería el algoritmo del vecino más cercano. Este algoritmo calcula para cada punto característico de una imagen los dos puntos

característicos más parecidos en la otra. Este algoritmo se aplica junto con el test de ratio en el que se descartan los emparejamientos en los que la distancia entre el más cercano y el punto a comparar y la distancia entre el segundo más cercano y el punto a comparar es demasiado similar. Para ello se establece un umbral de forma que un emparejamiento se considera válido si se cumple

$$dist1 \leq th \cdot dist2 \tag{2.1}$$

siendo $dist1$ la distancia con el vecino más cercano, $dist2$ la distancia con el segundo vecino más cercano y th el umbral establecido (normalmente 0,7).

Para calcular la distancia se utiliza la norma L1 para emparejar los descriptores SIFT y SURF (no binarios) y la distancia de Hamming para emparejar los descriptores binarios A-KAZE y ORB.

Estas técnicas se conocen como emparejamiento *pasivo*. Se calculan los puntos característicos en ambas imágenes y se emparejan. Existen otras técnicas como el algoritmo de Lucas-Kanade [14] basadas en el flujo óptico. En este algoritmo se detectan los puntos característicos en una imagen y se buscan estos puntos a lo largo de varias imágenes. Esto es lo que se conoce como emparejamiento *activo*. En estos emparejamientos, se buscan los puntos característicos detectados en la imagen anterior en la nueva imagen, sin necesidad de detectar puntos característicos en la nueva imagen.

En SLAM visual se utiliza emparejamiento activo para realizar un seguimiento de los puntos del mapa en la nueva imagen. Para ello, para cada punto del mapa estimado se utiliza la posición estimada de la cámara para predecir su posición en la nueva imagen y se buscan emparejamientos en una pequeña ventana cuyo centro está en la posición predicha de ese punto. Si en esta ventana hay un punto con un descriptor cuya distancia al descriptor del punto buscado esta por debajo de un umbral, entonces se emparejan. Si hay dos o más puntos cuyos descriptores están por debajo de este umbral, entonces se realiza el test de ratio (ecuación 2.1).

En este trabajo se van a evaluar diferentes detectores y extractores en ORB-SLAM2.

2.4. Interfaces visuales quirúrgicas

La visión por computador permiten que los ordenadores "vean" y "comprendan" los datos de las imágenes. Esta "comprensión" de las imágenes puede ayudar a la creación de sistemas con formas más naturales de interacción persona-ordenador. Estos sistemas pueden tener grandes aplicaciones en el campo de la medicina como el diagnóstico basado en imágenes o la asistencia en cirugía mínimamente invasiva.

Actualmente, los algoritmos de visión por computador pueden ayudar en el campo de la medicina de las siguientes formas:

- Mejora de las imágenes.

- Segmentación/detección de objetos como pólipos, piedras u otras estructuras anormales dentro del cuerpo humano así como los instrumentos [15].
- Extracción de puntos característicos para su posterior uso en tareas como SLAM o segmentación.
- Inserción de objetos virtuales o de información obtenida en la fase preoperatoria.

En las aplicaciones para la visualización avanzada de datos médicos, como los sistemas de realidad virtual o avanzada, se integran la visión por computador y la informática gráfica. Mientras que la visión por computador se encarga de extraer la información de las imágenes, la informática gráfica se encarga de crear objetos realistas y manipulables o interfaces para mostrar estructuras atómicas de interés o la información obtenida tras analizar la imagen.

Este tipo de aplicaciones se debe centrar en mejorar la precisión, la exactitud y la velocidad de cálculo, así como en reducir la cantidad de interacción manual. Además, las decisiones tomadas por estos sistemas deben poder ser sobreescritas por las decisiones tomadas por el cirujano.

Existen algunos proyectos en los que se ha desarrollado una sistema para controlar la interfaz del endoscopio realizando gestos con la mano derecha [16] o con la cara [17]. La principal desventaja es que estos sistemas pueden distraer al cirujano a la hora de realizar acciones quirúrgicas muy precisas.

Estos sistemas facilitan la interacción con la interfaz de endoscopio permitiendo aumentar el brillo, activar algunos modos de mejora de imagen o hacer zoom. Sin embargo, existen sistemas como los propuestos en [18], [19] y [20] que utilizan SLAM para estimar el mapa 3D del órgano que se está explorando y la posición del laparoscopio. La estimación de este mapa 3D permite la creación de objetos virtuales o la proyección de información obtenida en la fase pre-operatoria. En estos sistemas se puede ver el potencial que tiene el SLAM para el desarrollo de interfaces en el campo quirúrgico. Con una interfaz que use un sistema de SLAM es posible llevar un punto que se toca en una pantalla a un punto 3D en el mapa estimado. Además, permiten realizar anotaciones de realidad aumentada contextualizadas sin necesidad de ningún equipamiento adicional al endoscopio.

Aunque existen interfaces para manejar sistemas robóticos, como DaVinci [2] o MiroSurge [21], apenas existen interfaces que ayuden, mediante la adición de información visual, al cirujano a realizar operaciones endoscópicas.

[2] `https://www.davincisurgery.com/`

Capítulo 3

Método de seguimiento visual de la posición y la orientación

3.1. Modelo de la cámara

En este apartado se presenta la formulación del modelo de cámara *pinhole*, cámara estenopeica o cámara oscura. Este modelo es el modelo más simple que describe la relación matemática de la proyección de puntos del espacio 3D en un plano 2D de imagen (sección 3.1.1). En la sección 3.1.2 se abordará la formulación de las transformaciones rígidas mientras que en la sección 3.1.3 se considerarán las distorsiones de la lente.

3.1.1. Cámara *pinhole*

En la figura 3.1 se puede observar la proyección de un punto del espacio 3D $\boldsymbol{X}$, con coordenadas (X, Y, Z), en el plano de la imagen, $\boldsymbol{x}$ con coordenadas (u, v).

El origen del sistema de coordenadas está situado en el centro de la cámara, $\boldsymbol{c}$. La línea cuyo inicio está en el centro óptico, perpendicular al plano de la imagen se llama eje principal o rayo principal. El punto donde este eje corta al plano se llama punto principal, $\boldsymbol{p}$. El plano de la imagen está situado a una distancia f (distancia focal) en el eje ortogonal $\boldsymbol{Z}$ (o eje principal) del centro de la cámara. En realidad, el plano de la imagen se encuentra detrás del centro óptico de la cámara ($z = -f$). Sin embargo, por simplicidad se asume delante ($z = f$). Considerando la semejanza de triángulos puede establecerse la siguiente regla de tres (Teorema de Tales):

$$f \rightarrow Z \tag{3.1}$$

$$u \rightarrow X \tag{3.2}$$

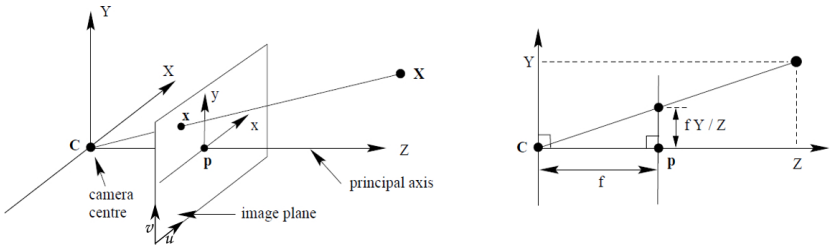


Figura 3.1: Modelo cámara *pinhole*[3]

Despejando la regla de tres quedaría:

$$u = \frac{X}{Z} f \tag{3.3}$$

Análogamente para la componente Y:

$$v = \frac{Y}{Z} f \tag{3.4}$$

Con este modelo un punto en el espacio 3D con coordenadas (X, Y, Z) se convertiría en $(\frac{X}{Z}f, \frac{Y}{Z}f, f)$. Ignorando la última coordenada se obtiene las coordenadas 2D en la imagen $(\frac{X}{Z}f, \frac{Y}{Z}f)$.

Asumiendo que los puntos están representados en coordenadas homogéneas (un punto 2D se representa con una terna $\tilde{\boldsymbol{x}} \propto (x, y, w)$), la proyección puede expresarse en términos matriciales de la siguiente forma:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \propto \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} fX_{\text{cam}} \\ fY_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix} = \boldsymbol{K}\boldsymbol{X} \tag{3.5}$$

En teoría, el origen de las coordenadas en el plano de la imagen se supone que está en el punto principal. Esto puede no ser cierto en la práctica, en ese caso la proyección se expresaría de la siguiente forma:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \propto \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} fX_{\text{cam}} + Z_{\text{cam}}p_x \\ fY_{\text{cam}} + Z_{\text{cam}}p_y \\ Z_{\text{cam}} \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix} = \boldsymbol{K}\boldsymbol{X} \tag{3.6}$$

donde p_x, p_y serían las coordenadas del punto principal $\boldsymbol{p}$. La matriz $\boldsymbol{K}$ es conocida como la matriz intrínseca y generalmente se expresa de la siguiente forma:

$$\boldsymbol{K} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

donde:

- $\alpha_x = \frac{f}{d_x}$ es la distancia focal horizontal en píxeles.

[3]Imagen obtenida de `https://hedivision.github.io/Pinhole.html`

- $\alpha_y = \frac{f}{d_y}$ es la distancia focal vertical en píxeles.
- d_x, d_y son el tamaño en píxeles de u e v.
- p_x, p_y las coordenadas del punto principal expresadas en píxeles.
- s es el sesgo o *skew*, normalmente 0.

3.1.2. Transformaciones rígidas

En la formulación se ha referido a las coordenadas del punto en el espacio 3D con el sufijo *cam*. Esto es para enfatizar que estas coordenadas son en referencia al sistema de coordenadas de la cámara. Para poder trabajar con puntos de diferentes cámaras, estos han de hacer referencia a un sistema de coordenadas global. Estos sistemas están relacionados mediante matrices de rotación y traslación (figura 3.2). Si $\tilde{\boldsymbol{X}}$ tiene coordenadas $(X, Y, Z, 1)$ en el sistema de referencia global, para obtener las coordenadas en el sistema de referencia de la cámara se ha de aplicar la siguiente fórmula:

$$\boldsymbol{X}_{\text{cam}} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{X}} \tag{3.8}$$

donde $\boldsymbol{R}$ es la matriz de rotación y tiene una dimensión de 3×3 y $\boldsymbol{t}$ es el vector de traslación y tiene una dimensión de 3×1.

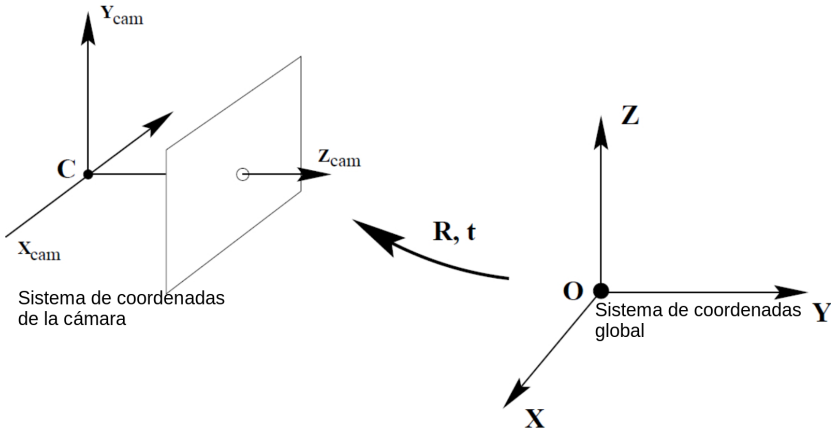


Figura 3.2: Sistemas de coordenadas[4].

La rotación y la traslación son conocidos como los parámetros extrínsecos y la matriz $\boldsymbol{K}$ y los coeficientes de distorsión como los parámetros intrínsecos. Juntando todo se obtiene la matriz de

[4]Imagen obtenida de `https://hedivision.github.io/images/rt.jpg`

proyección $\boldsymbol{P}$:

$$\boldsymbol{P} = \boldsymbol{K} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \tag{3.9}$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{P}\tilde{\boldsymbol{X}} \tag{3.10}$$

3.1.3. Distorsión de la lente

La distorsión radial es la aberración geométrica más importante de la lente y se produce cuando los rayos de luz se doblan más cerca de los bordes de una lente que en su centro óptico (figura 3.3). Esto produce que el rayo proyectado desde el centro de la cámara que pasa por el punto 2D distorsionado no intersecte con el punto correspondiente en el espacio 3D. Cuanto más pequeña es la lente, mayor es la distorsión y más grande el ángulo de apertura.

Figura 3.3: Tipos de distorsión de la lente [5].

Para obtener las coordenadas no distorsionadas de un punto es necesario obtener los coeficientes de distorsión de la cámara. Para ello se han de tomar imágenes de un patrón de tablero de ajedrez (como el patrón de la imagen de la figura 3.3) desde diferentes ángulos y puntos de vista [22].

Siendo $\boldsymbol{x}_d = (u_d, v_d)$ las coordenadas distorsionadas de un punto, para obtener las coordenadas no distorsionadas se ha de aplicar la siguiente fórmula:

$$u_d = u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3.11}$$

$$v_d = v(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3.12}$$

$$r^2 = u^2 + v^2 \tag{3.13}$$

donde k_1, k_2, k_3 son los coeficientes de distorsión y (u, v) son las coordenadas no distorsionadas [23].

3.2. Puntos naturales de interés

En visión por computador un punto de interés es un punto en la imagen que contiene información relevante. En el caso de la localización geométrica, estos puntos suelen ser esquinas, ya que son puntos

[5] Imagen obtenida de `https://docs.opencv.org/2.4/_images/distortion_examples.png`

que tienen cambios en las dos direcciones o ejes a diferencia de los bordes que solo tienen cambio en una dirección y un punto en un borde es igual que cualquier otro punto en ese mismo borde (figura 3.4).

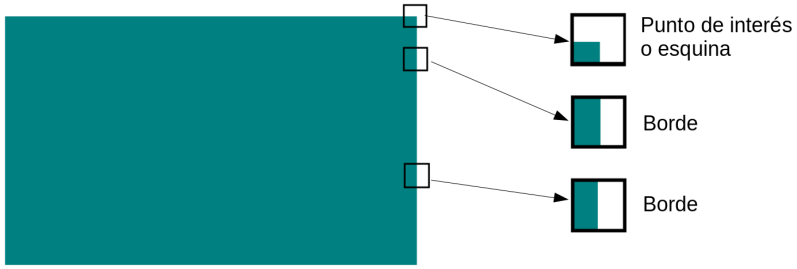


Figura 3.4: Diferencia entre punto de interés o esquina y borde.

3.2.1. Detectores y descriptores

Existen diferentes algoritmos de detección de puntos de interés. Estos algoritmos, llamados *detectores*, se basan en diferentes métodos para obtener puntos de interés *invariantes* a la iluminación, a la escala, a la rotación, a la posición en la imagen, a los cambios de afín o a los cambios de escala.

Sin embargo, para poder comparar los puntos de interés de dos imágenes es necesario que estos puntos sean descritos mediante un *descriptor*. Este descriptor debe ser capaz de describir el punto de interés teniendo invarianza a los diferentes cambios mencionados pero sin embargo *variar* con respecto a la textura y permitiendo comparar dos descriptores de la forma más eficiente posible. En este trabajo se va a utilizar la notación *detector-descriptor* para referirse a cada una de las diferentes combinaciones.

Como ya se ha mencionado en el apartado 2.3, existen diferentes detectores y descriptores. ORB-SLAM2 utiliza el detector y descriptor ORB (FAST-ORB), es decir usa el detector FAST modificado (invariante a la escala) y describe estos puntos usando el descriptor BRIEF modificado (invariante a la rotación, ORB). Estos descriptores son descriptores binarios por lo que su tiempo de cómputo y emparejamiento es bajo (aproximadamente 40 ms).

La principal desventaja de utilizar puntos FAST es que, aunque su tiempo de cómputo sea bajo, tienen baja repetibilidad, es decir, un punto de interés no siempre es detectado a lo largo de varias imágenes [24] [10].

Uno de los problemas que se encuentra a la hora de intentar detectar puntos de interés en las imágenes capturadas por el endoscopio es la dificultad de encontrar puntos de interés fiables en el interior del cuerpo debido a la falta de texturas y de esquinas. Debido a esto, y a que FAST considera como punto de interés un punto rodeado por un círculo de 16 píxeles de los cuales n píxeles consecutivos son más claros o más oscuros que el punto que está siendo evaluado [12], se detectan pocos puntos

FAST en las imágenes médicas y algunos de los puntos detectados pueden ser debidos al ruido que tienen las imágenes de este tipo de cámaras. En la figura 3.5 se puede observar un punto que sería considerado como punto de interés FAST ya que de los 16 puntos que rodean al punto $\boldsymbol{p}$, los puntos del 1 al 4 y del 10 al 16 son más oscuros que el punto $\boldsymbol{p}$.

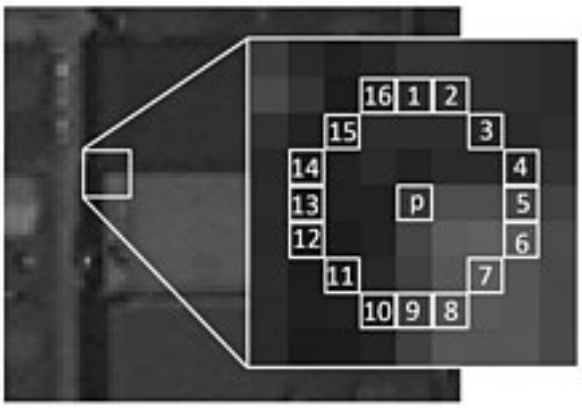


Figura 3.5: Punto de interés FAST [6].

En las figuras 3.6 y 3.7 se pueden observar los puntos detectados en dos entornos bastante diferentes. En 3.7 se pueden observar los puntos detectados en una secuencia del dataset *EuRoC* [7] que corresponde a una habitación con objetos dispersos en ella. En esta imagen se puede ver que la mayoría de los puntos se detectan en las esquinas. En la figura 3.6 se pueden observar los puntos detectados en el interior de una vejiga. En este caso se puede ver que los puntos se detectan en intersecciones de venas o en venas. En estas imágenes también se puede observar cómo en la vejiga se detectan menos puntos que en la habitación de EuRoC, que la calidad de las imágenes del endoscopio tienen más ruido, y que en las imágenes del endoscopio se trabaja muy cerca de las paredes de los órganos, por lo que se tienen una visión de un área mucho menor.

Debido a la baja repetibilidad de los puntos FAST y a la dificultad para encontrar puntos fiables en el interior del cuerpo, en este trabajo se han evaluado los siguientes detectores: FAST, A-KAZE y Shi-Tomasi con los descriptores ORB y A-KAZE. Como ambos descriptores son binarios, se utiliza la distancia de Hamming para emparejar puntos.

3.2.2. A-KAZE

En esta sección se va a explicar con más detalle el detector y descriptor *A-KAZE*[10], es decir, de la combinación A-KAZE - A-KAZE ya que es un método relativamente nuevo (2013) con el que se han

[6] Imagen de Jingjin Huang, Guoqing Zhou, Xiang Zhou and Rongting Zhang - `https://www.mdpi.com/1424-8220/18/4/1014,CCBY4.0`, `https://commons.wikimedia.org/w/index.php?curid=87399771`

[7] `https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets`

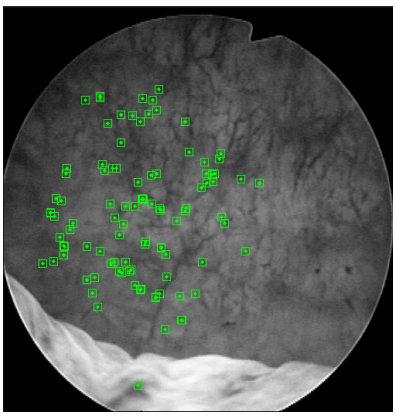


Figura 3.6: Puntos detectados por ORB-SLAM2 en una vejiga.

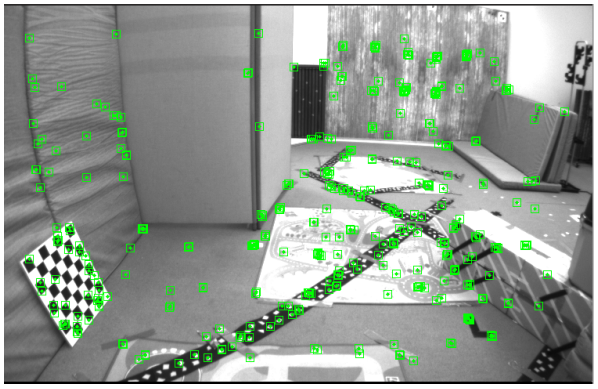


Figura 3.7: Puntos detectados por ORB-SLAM2 en una habitación del dataset EuRoC.

obtenido buenos resultados. Este método explota el espacio de escala no lineal para extraer los puntos característicos de una imagen.

El primer paso que se realiza para la detección de puntos de interés es la construcción del espacio de escala no lineal. Los niveles de espacio de escala se construyen resolviendo numéricamente la ecuación diferencial parcial (Ec. 3.14) de manera iterativa utilizando el esquema de Difusión Explícita Rápida (FED, *Fast Explicit Diffusion*) con pasos de tiempo variable τ_i(Ec. 3.15).

$$\frac{\partial \boldsymbol{L}}{\partial t} = \text{div}(g(|\nabla \boldsymbol{L}_\sigma|) \cdot \nabla \boldsymbol{L}) \tag{3.14}$$

$$\boldsymbol{L}^{i+1} = (\boldsymbol{I} + \tau_i \boldsymbol{A}(\boldsymbol{L}^i))\boldsymbol{L}^i \tag{3.15}$$

La función $g(|\nabla \boldsymbol{L}_\sigma|)$ es la función de conductividad o de difusión, $\boldsymbol{L}$ es la luminancia de la imagen, $\boldsymbol{L}^i$ es la solución en nivel de evolución i y $\boldsymbol{A}(\boldsymbol{L}_i)$ es la matriz que representa la difusión de la imagen

[25] e $\boldsymbol{I}$ la matriz identidad. La función de difusión elegida es g_2 [26]:

$$g_2 = \frac{1}{a + \frac{|\nabla \boldsymbol{L}\sigma|^2}{\lambda^2}} \tag{3.16}$$

donde el parámetro λ es el factor de contraste que controla el nivel de difusión. Determina qué bordes deben ser mejorados o mantenidos y cuáles deben ser anulados. Valores bajos conservarán la mayoría de los bordes mientras que valores altos solo conservarán los bordes con gradiente más fuerte. Esta función favorece regiones amplias sobre otras más pequeñas.

Considerando la estimación a priori $\boldsymbol{L}^{i+1,0} = \boldsymbol{L}^i$, un ciclo de FED con n pasos de tamaño variable τ_j se obtiene como:

$$\boldsymbol{L}^{i+1,j+1} = (\boldsymbol{I} + \tau_i \boldsymbol{A}(\boldsymbol{L}^i))\boldsymbol{L}^{i+1,j}, \quad j = 0, ..., n-1 \tag{3.17}$$

Para construir el espacio de escala no lineal en primer lugar, es necesario definir un conjunto de tiempos de evolución a partir de los cuales se pueda construir este espacio. El espacio de escala está discretizado en una serie de O octavas o niveles y S subniveles. El conjunto de octavas y subniveles se identifican por un índice discreto de octava o y un índice discreto de subnivel s. Los índices de octava y de subnivel se asignan a su escala σ (píxeles) correspondiente mediante la siguiente ecuación:

$$\sigma_i(o, s) = 2^{\frac{o+s}{S}}, \quad o \in [0...O-1], \quad s \in [0...S-1], \quad i \in [0...M] \tag{3.18}$$

donde M es el número de imágenes filtradas (pirámide). Ahora, es necesario convertir el conjunto de niveles discretos de escala en unidades de píxeles σ_i a unidades de tiempo, ya que el filtrado de difusión no lineal opera en unidades de tiempo. Se utiliza el mapeo $\sigma_i \rightarrow t_i$ descrito en [27] para convertir de unidades de píxeles a unidades de tiempo:

$$t_i = \frac{1}{2}\sigma_i^2, \quad i = 0...M \tag{3.19}$$

Además, sobre la imagen de entrada se puede aplicar un suavizado de imagen con un filtro Gaussiano de desviación estándar σ_0 para reducir el ruido y los posibles artefactos. A partir de la imagen de entrada suavizada se calcula el contrastante λ (ecuación 3.16) de forma automática como el percentil 70 % del histograma de gradiente.

Se considera que cada ciclo exterior de FED cubre un tiempo de ciclo $T = t_i + 1 - t_i$. Una vez que se llega al último subnivel de cada octava o nivel, se reduce la imagen en un factor de 2 utilizando la máscara de suavizado $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ y utilizando esa imagen reducida como imagen de partida para el siguiente ciclo de FED en la siguiente octava. Tras reducir la imagen, es necesario modificar el parámetro de contraste λ. La máscara de suavizado reduce el contraste de un borde ideal en ese paso en un 25 %, y por lo tanto el parámetro de contraste necesita ser multiplicado por 0,75. Para una mejor

comprensión, se resume la aproximación de FED piramidal para el filtrado de difusión no lineal en Alg. 1. De la misma manera, las iteraciones del ciclo interno de FED se describen en el Alg. 2.

Algorithm 1: Aproximación de FED piramidal para el filtrado de difusión no lineal

Input : Imagen $\boldsymbol{L}^0$, parámetro de contraste λ y conjunto de tiempos de evolución t_i
Output: Conjunto de imágenes filtradas (pirámide)
for $i = 0 \rightarrow M - 1$ **do**
  1. Calcular la matriz de difusividad $\boldsymbol{A}(\boldsymbol{L}^i)$
  2. Establecer el tiempo de ciclo exterior de FED $T = t_{i+1} - t_i$
  3. Calcular el número de pasos internos de FED n
  4. Calcular el tamaño de los pasos de tiempo τ_j
  5. Establecer $\boldsymbol{L}^{i+1,0} = \boldsymbol{L}^i$
  $\boldsymbol{L}^{i+1} = \boldsymbol{FEDCYCLE}(\boldsymbol{L}^{i+1,0}, \boldsymbol{A}(\boldsymbol{L}^i), \tau_j)$
  if $o_{i+1} > o_i$ **then**
    Reducir $\boldsymbol{L}^{i+1}$ con la máscara $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$
    $\lambda = \lambda \cdot 0{,}75$
  end
end

Algorithm 2: Ciclo interno FED

Function `FEDCYCLE`($\boldsymbol{L}^{i+1,0}$, $\boldsymbol{A}(\boldsymbol{L}^i)$, τ_j)**:**
  for $j = 0 \rightarrow n - 1$ **do**
    $\boldsymbol{L}^{i+1,j+1} = (\boldsymbol{I} + \tau_j \boldsymbol{A}(\boldsymbol{L}^i))\boldsymbol{L}^{i+1,j}$
  return $\boldsymbol{L}^{i+1,n}$

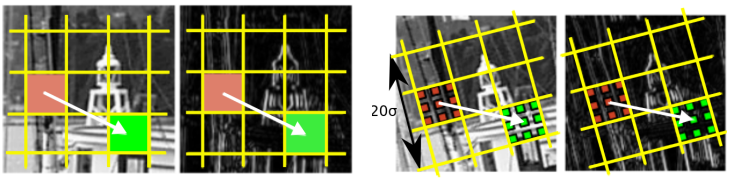
Para detectar los puntos, se calcula el determinante del Hessiano para cada una de las imágenes filtradas $\boldsymbol{L}_i$ en el espacio de escala no lineal. El conjunto de operadores diferenciales de multiescala se normalizan con respecto a la escala, utilizando un factor de escala normalizado que tiene en cuenta la octava de cada imagen en el espacio de escala no lineal, es decir, $\sigma_{i,\text{norm}} = \sigma_i / 2^{oi}$, y

$$\boldsymbol{L}^i_{\text{Hessiano}} = \sigma^2_{i,\text{norm}}(\boldsymbol{L}^i_{xx}\boldsymbol{L}^i_{yy} - \boldsymbol{L}^i_{xy}\boldsymbol{L}^i xy) \tag{3.20}$$

Para calcular las derivadas de segundo orden ($\boldsymbol{L}_{xx}$, $\boldsymbol{L}_{yy}$) se utilizan filtros de Scharr concatenados con un tamaño de paso de $\sigma_{i,\text{norm}}$.

Para detectar los puntos característicos, primero se buscan máximos de la respuesta del detector en la misma escala. Para ello en cada nivel de evolución i, se comprueba que la respuesta del detector es mayor que un umbral predefinido y es un máximo en una ventana de 3×3 píxeles. Luego, para cada uno de estos máximos se comprueba si es un máximo con respecto a los niveles $i+1$ e $i-1$, respectivamente, el nivel de arriba y el nivel de abajo en una ventana de tamaño $\sigma_i \times \sigma_i$ píxeles. La posición 2D del punto característico es estimada con una precisión de sub-pixel ajustando una función 2D cuadrática a la respuesta del determinante del Hessiano en un vecindario de 3×3 píxeles y encontrando su máximo.

Por último para describir los puntos característicos obtenidos se utiliza el descriptor M-LDB (*Modified-Local Difference Binary*) que explota la información de gradiente y de intensidad del espacio de escala no lineal. El descriptor LDB [28] sigue el mismo principio que el descriptor BRIEF [13], pero utilizando pruebas binarias entre la media de las áreas en lugar de píxeles individuales para una mayor robustez. Además de los valores de intensidad, se utiliza la media de las derivadas horizontales y verticales de las áreas comparadas, lo que resulta en 3 bits por comparación. LDB propone utilizar varias cuadrículas, dividiendo el parche en cuadrículas de 2×2, 3×3, 4×4, etc. (figura 3.8(a)). Los promedios de esas subdivisiones son muy rápidos de calcular utilizando imágenes integrales, sin embargo, cuando se considera la rotación de los puntos característicos no se pueden utilizar imágenes integrales. La invariancia de la rotación se obtiene estimando la orientación principal del punto característicos, y rotando la cuadrícula de LDB en consecuencia. En lugar de utilizar el promedio de todos los píxeles dentro de cada subdivisión de la cuadrícula, M-LDB submuestrea la cuadrícula dependiendo de la escala haciendo que el descriptor sea robusto a los cambios de escala (figura 3.8(b)). M-LDB utiliza las derivadas calculadas en el paso de detección de puntos característicos, reduciendo el número de operaciones necesarias para construir el descriptor.



(a) LDB. (b) M-LDB.

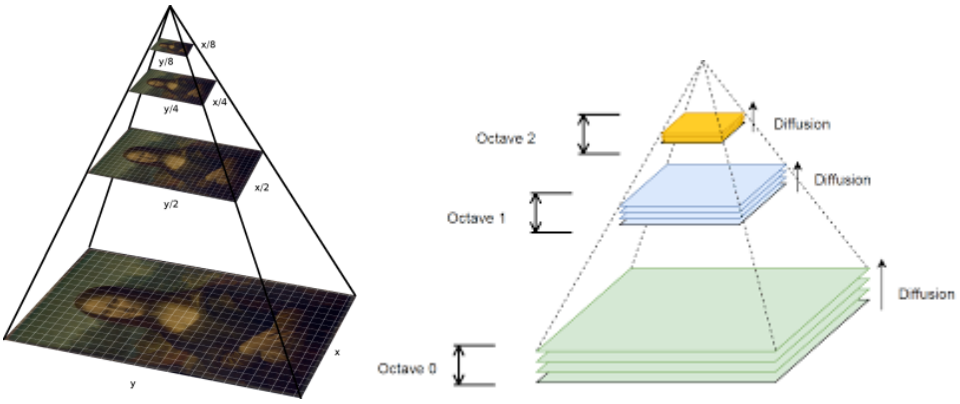
Figura 3.8: Diferencia entre los descriptores LDB y M-LDB [10].

3.2.3. Representación piramidal de imágenes

Para hacer estos detectores invariantes a la escala, se calcula una pirámide compuesta por la imagen en diferentes escalas y se detectan los puntos en todos los niveles. En ORB-SLAM2 se calcula la pirámide de diferentes escalas mientras que en A-KAZE se calcula un espacio de escala no lineal (*Non-linear Scale Space*, NLS). Para hacer el detector de Shi-Tomasi invariante a la escala, este se utiliza tanto en la representación piramidal como en el espacio de escala no lineal.

La principal diferencia entre estas representaciones es que en la pirámide se reduce el tamaño de la imagen en un factor de escala fijo (normalmente 1,2). En la pirámide Gaussiana además se aplica el filtro Gaussiano suavizando toda la imagen, tanto los detalles como el ruido, mientras que en el espacio de escala no lineal el desenfoque se adapta localmente a los datos de la imagen, suavizando el ruido y manteniendo los detalles o las zonas en las que se pueden encontrar puntos característicos.

Estas diferencias se puede observar en las figuras 3.9 y 3.10.



(a) Representación piramidal de una imagen [9].

(b) Representación piramidal del espacio no lineal [29].

Figura 3.9: Representaciones piramidales.

Figura 3.10: Diferencia entre la pirámide Gaussiana (arriba) y espacio de escala no lineal (abajo) [10].

3.3. Pre-procesado de imágenes

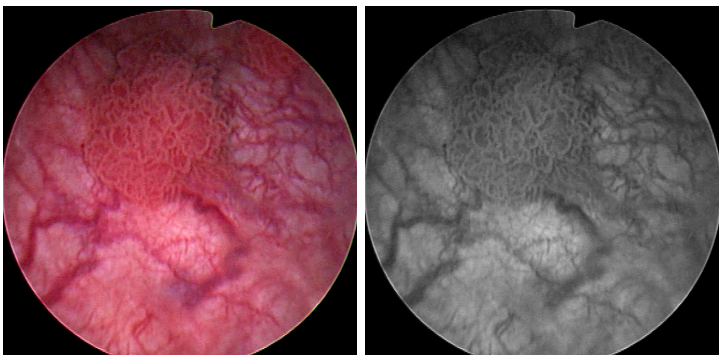
Uno de los problemas que se encuentran a la hora de detectar y describir los puntos característicos de una imagen es que las imágenes tomadas por el endoscopio tienen ruido y que hay cambios bruscos de iluminación. Por estos motivos, las imágenes capturadas por el endoscopio son procesadas para que se pueda extraer el mayor número posible de puntos característicos visuales fiables.

3.3.1. Selección de banda de color

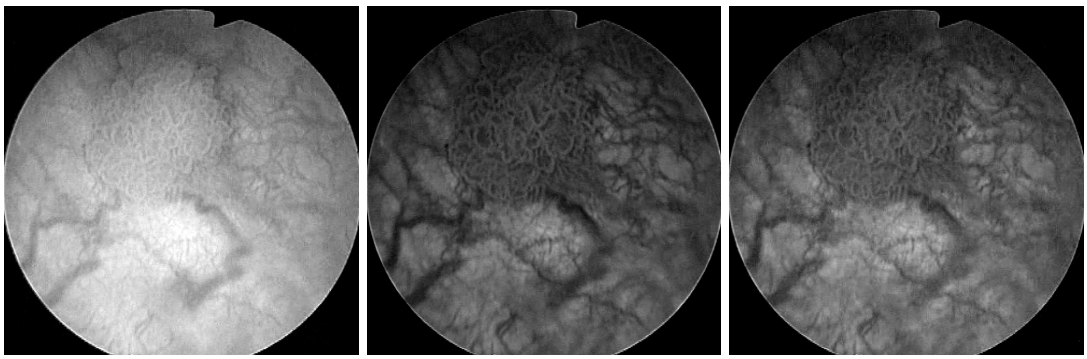
Lo primero que se ha de tener en cuenta es que para poder usar los detectores mencionados sobre la imagen, ésta ha de estar en blanco y negro o en escala de grises (imágenes con un único canal). Una

[9]Imagen obtenida de `http://coding-guru.com/image-pyramid-expanding-image-2/`

imagen a color está compuesta por tres canales: rojo (R: *red*), verde (G: *green*) y azul (B: *blue*). Para pasar una imagen de color a blanco y negro o escalas de grises, se hace la media de los tres canales, obteniendo una imagen de un único canal. Otra opción para obtener una imagen de un único canal es elegir uno de los tres canales mencionados. Como se puede ver en la figura 3.11, el canal verde es el canal que mejor conserva las características visuales y el contraste. El canal azul es otro candidato pero este canal conserva más ruido que el verde. Debido a esto, sólo se utiliza el canal verde.



(a) Imagen original. (b) Escala de grises.



(c) Canal rojo. (d) Canal verde. (e) Canal azul.

Figura 3.11: Descomposición de la imagen RGB.

Otro de los problemas es la presencia de reflejos en algunas imágenes de la secuencia. La presencia de reflejos causa que se detecten puntos en éstos y que se emparejen con otros puntos detectados en los reflejos y por lo tanto que se añadan al mapa puntos erróneos ya que no se ven los mismos reflejos si la cámara se mueve (la luz artificial está fijada en el endoscopio por lo que se mueve junto con éste). Para lidiar con este problema, se ha creado una máscara para detectar las zonas en las que hay reflejos e ignorar los puntos detectados en esa zona. Para ello se sigue el siguiente proceso (figura 3.12):

1. Convertir la imagen a HSV (*Hue, Saturation, Value*, en español, tonalidad, saturación, brillo o valor).

2. Extraer el canal que contiene información sobre la saturación (canal S).

3. Aplicar sobre este canal un filtrado utilizando el método de Otsu's [30] para calcular de forma automática un umbral a partir del histograma de la imagen. Una vez calculado el umbral, se convierten los píxeles cuyo valor esté por encima de este umbral a 0 (negro) y los píxeles cuyo valor esté por debajo de este umbral a 255 (blanco).

4. Dilatar la máscara para que no se detecten puntos en las zonas que rodean al reflejo.

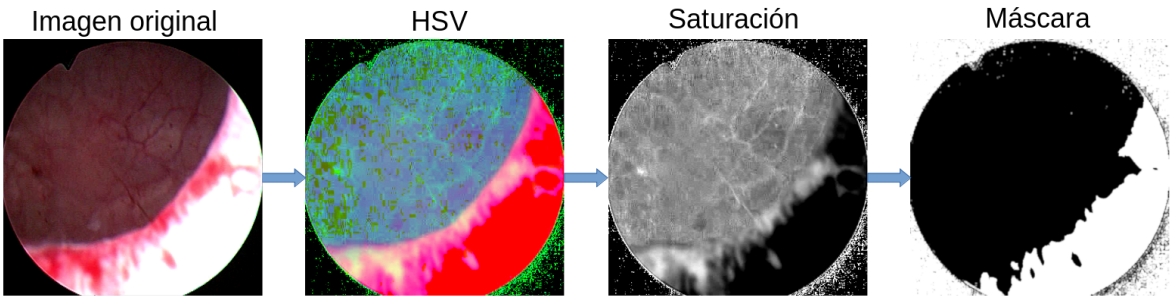
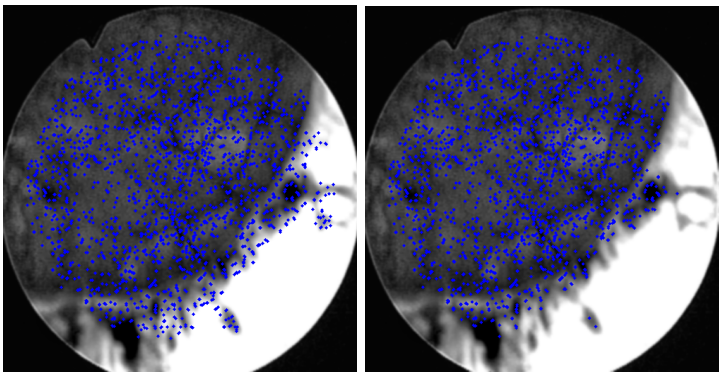


Figura 3.12: Proceso seguido para crear la máscara para detectar reflejos.

Esta máscara se utiliza para que no se detecten puntos en los reflejos. En la figura 3.13 se puede observar cómo al usar la máscara no se detectan puntos en las zonas en las que hay reflejos.



(a) Puntos detectados sin usar la máscara.

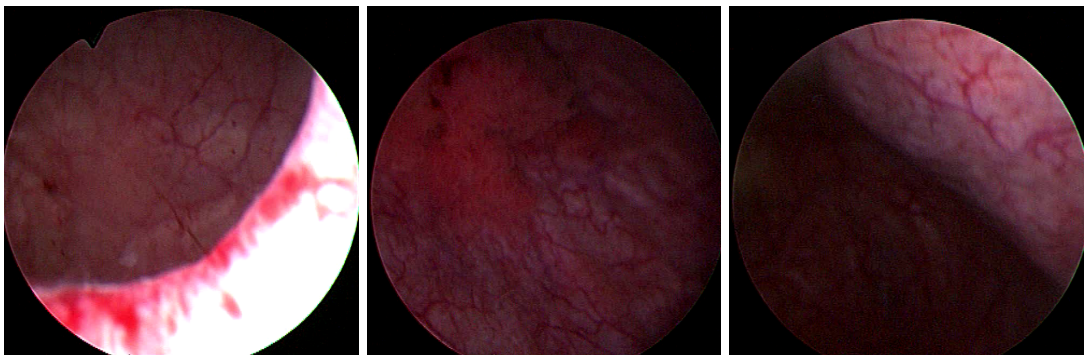
(b) Puntos detectados usando la máscara.

Figura 3.13: Diferencia en la detección de puntos al usar la máscara para evitar detectar las características visuales en los reflejos.

3.3.2. CLAHE

Como ya se ha mencionado uno de los retos de las secuencias médicas son los cambios bruscos de iluminación. En nuestras secuencias se puede encontrar imágenes con reflejos (figura 3.14(a)), imágenes

con baja iluminación (figura 3.14(b)) o imágenes en las que una parte de la imagen tiene buena iluminación pero la otra tenga una baja iluminación (figura 3.14(c)). Debido a esto se ha decidido utilizar el método CLAHE (*Contrast Limited Adaptive Histogram Equalization*) [31].



(a) Reflejos. (b) Baja iluminación. (c) Parte de la imagen con buena iluminación y parte de la imagen con iluminación baja.

Figura 3.14: Problemas encontrados con la iluminación.

Para entender CLAHE primero hay que entender el concepto de la ecualización del histograma de valores (brillo) de la imagen. Considérese una imagen cuyos valores de píxeles están limitados a un rango específico de valores. Por ejemplo, en una imagen muy brillante los valores de los píxeles estarán concentrados en un rango alto. Sin embargo, una imagen con mayor contraste tendrá diferentes valores de píxeles en todas las regiones de la imagen. Teniendo en cuenta esto, una técnica que se suele utilizar para aumentar el contraste de una imagen es estirar su histograma hacia cualquiera de los extremos (figura 3.15). Esto es lo que hace la ecualización del histograma aunque, en la práctica, tras aplicar la ecualización no se obtiene un histograma completamente plano.

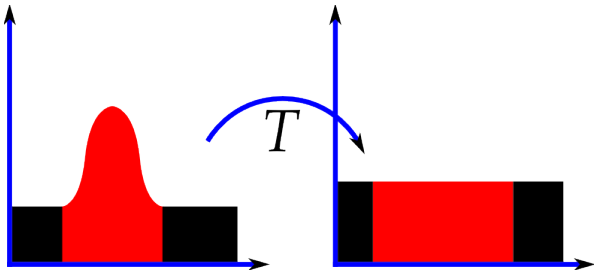


Figura 3.15: Ecualización del histograma [10].

La desventaja de la ecualización del histograma es que considera el contraste global de la imagen, por lo que se obtienen buenos resultados con imágenes que son muy brillantes o muy oscuras pero no

[10]Imagen obtenida de `https://en.wikipedia.org/wiki/Histogram_equalization`

se obtienen buenos resultados en imágenes que tienen partes con diferente iluminación.

Para resolver este problema, se utiliza la ecualización de histograma adaptativo (AHE). En este caso, la imagen se divide en un número establecido de pequeños bloques cuadrados no superpuestos llamados "tiles" y se aplica la ecualización del histograma a cada uno de ellos. El problema de AHE es que si hay ruido en la imagen y un color predominante, el ruido se amplifica. Para evitar esto se aplica la limitación de contraste (CLAHE). Si cualquier valor del histograma está por encima del límite de acumulación o contraste establecido, esos píxeles se recortan y se distribuyen uniformemente en las otras cajas del histograma antes de aplicar la ecualización (figura 3.16). De esta forma se limita el aumento de contraste y, por tanto, la amplificación del ruido. Por último, cómo la ecualización se aplica a cada "tile" de forma independiente, se aplica interpolación en brillo para corregir las inconsistencias entre los bordes de los "tiles" y evitar que se vean las "tiles" en la imagen mejorada por CLAHE.

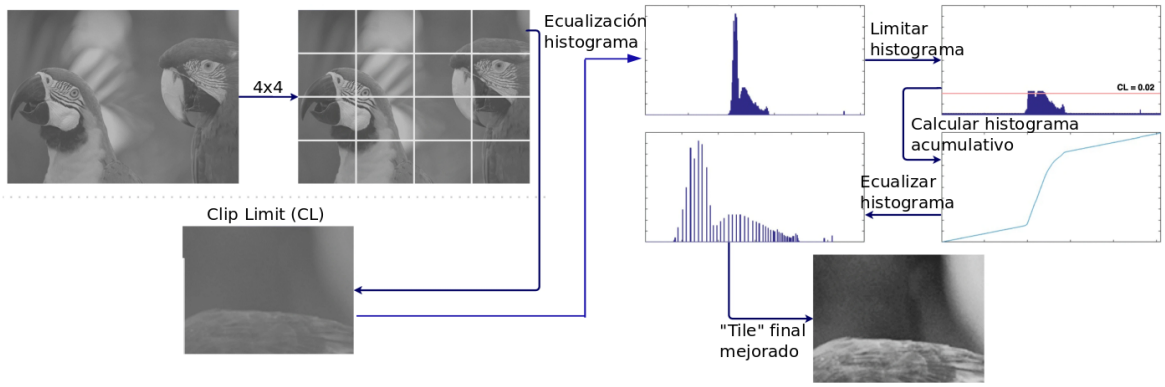
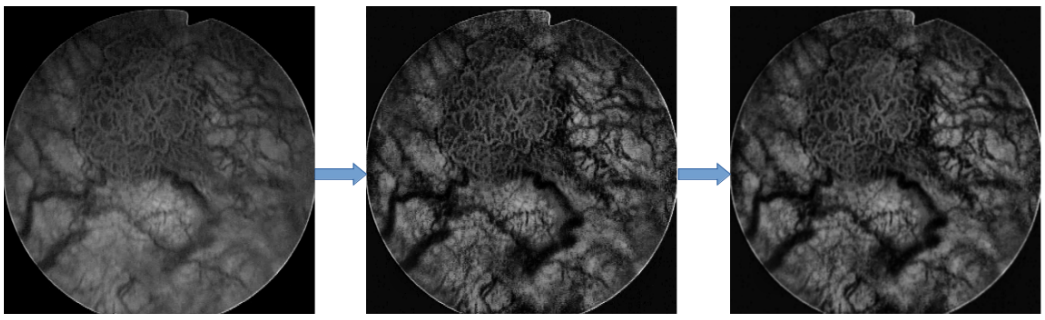
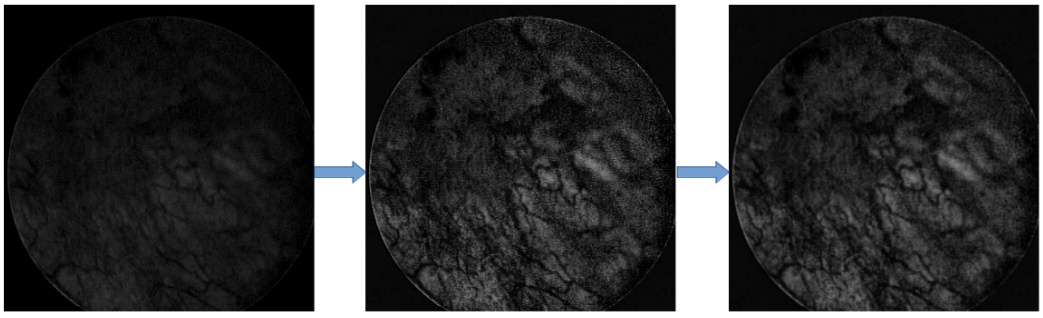


Figura 3.16: CLAHE [32].

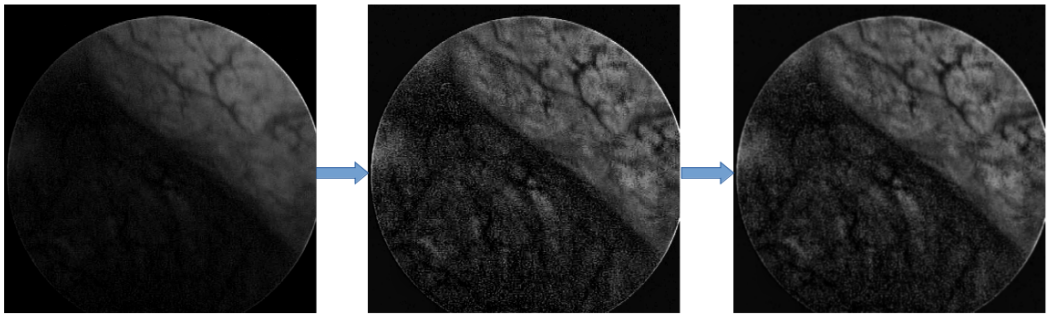
Para aplicar CLAHE sobre las imágenes de color tomadas por el endoscopio, primero se transforma la imagen al espacio de color Lab. El canal L de esta imagen representa la iluminación por lo que se aplica CLAHE sobre este canal. Una vez se haya aplicado CLAHE, se vuelven a unir los canales L, a y b y se convierte la imagen del espacio de color Lab al espacio de color BGR. De esta imagen se selecciona únicamente el canal G que representa el canal verde de la imagen. Siguiendo este proceso se obtienen mejores resultados que al aplicar CLAHE directamente sobre el canal verde. En la figura 3.17 se puede observar que tras aplicar CLAHE, la imagen aún tiene mucho ruido. Debido a esto, se aplica un filtro Gaussiano para suavizar la imagen y eliminar el ruido. En esta figura también se puede observar cómo al usar CLAHE aumenta el contraste de la imagen.



(a) Imagen normal.



(b) Imagen oscura.



(c) Imagen de iluminación variada.

Figura 3.17: Resultado de aplicar CLAHE + filtro Gaussiano.

3.4. ORB-SLAM2

ORB-SLAM2 [3] es un software desarrollado por la Universidad de Zaragoza que actualmente se utiliza en diferentes proyectos en diferentes partes del mundo. ORB-SLAM2 estima la posición y orientación (*pose*) de la cámara y el mapa 3D del entorno que la rodea a partir de la información

obtenida de las imágenes capturadas por ésta.

ORB-SLAM2 está compuesto por tres hilos de ejecución (figura 3.18). Estos hilos se ejecutan de forma paralela y cada uno se encarga de una tarea diferente:

- *Tracking*: encargado de estimar la *pose* de la cámara en cada imagen (*frame*).
- *Local Mapping*: encargado de añadir puntos al mapa y de optimizar la *pose* de la cámara y el mapa.
- *Loop Closing*: encargado de detectar y cerrar bucles.

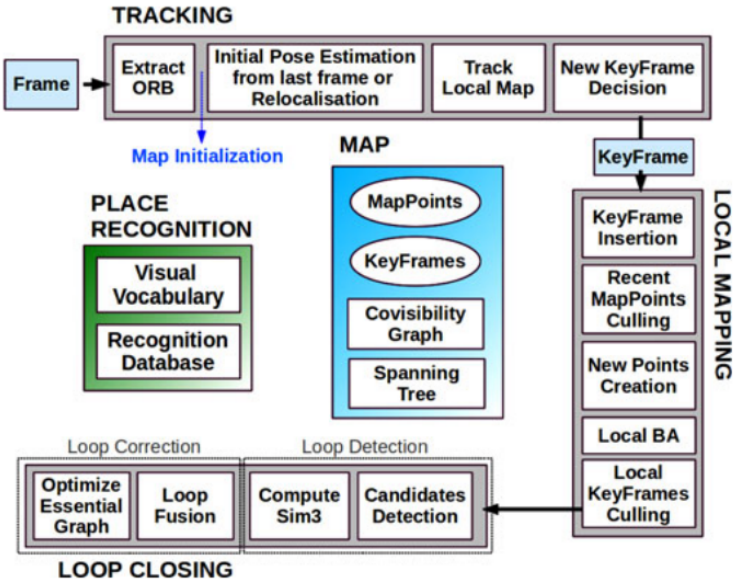


Figura 3.18: Estructura de ORB-SLAM2 [3].

Una de las características de este sistema es que a largo plazo, no trabaja con todas las imágenes (*frames*) capturadas por la cámara sino que sólo trabaja con aquellas que considera importantes, denominados *keyframes*. Un *frame* se considera *keyframe* si contiene suficiente información nueva con respecto al resto de *keyframes* almacenados. Además, estos *keyframes* están relacionados mediante un grafo de covisibilidad. En este grafo cada nodo es un *keyframe* y el "peso" de las aristas del grafo representa el número de puntos que ambos *keyframes* tienen en común.

Otra característica de este sistema es que utiliza descriptores ORB. Por cada nueva imagen o *frame* se extraen los puntos de interés FAST y se describen mediante el descriptor ORB. Éstos son emparejadas con los descriptores ORB de los diferentes *keyframes* almacenados. Una vez emparejada una característica visual o punto de interés, se triangula y se añade al mapa. En la sección 3.4.1 se entrará en más detalle en este procedimiento.

3.4.1. Seguimiento de puntos de interés

Cada vez que se captura una nueva imagen, el primer paso es extraer los puntos FAST en las diferentes escalas, describir estos puntos con descriptores ORB y desdistorsionar o rectificar la posición de estos puntos (sección 3.1.3). ORB-SLAM2 no rectifica todos los puntos de la imagen si no que sólo rectifica los puntos característicos. Si el seguimiento fue exitoso en la última imagen, se utiliza un modelo de aceleración constante para predecir la nueva posición y orientación de la cámara realizando una *búsqueda guiada* de los puntos observados en la última imagen usando una *ventana de búsqueda* por cada punto observado en la imagen anterior cuyo centro está en la posición predicha del punto que se está intentando emparejar. Si no se han emparejado puntos suficientes, se aumenta el tamaño de esta ventana permitiendo realizar una búsqueda más exhaustiva. Si se han encontrado suficientes puntos, se realiza una optimización de la posición y orientación. En caso contrario, el sistema pasa a estar en el estado "perdido" y se realiza relocalización en cada nueva imagen (sección 3.4.3).

Una vez obtenida la estimación de la posición y la orientación de la cámara y un conjunto inicial de emparejamientos, se proyectan los puntos del mapa local en la imagen y se buscan más correspondencias de puntos del mapa. El mapa local contiene un conjunto de *keyframes* K_1 que comparte puntos con la imagen actual y un conjunto de *keyframes* K_2 que contiene los vecinos de los *keyframes* K_1 en el grafo de covisibilidad. También tiene un conjunto de *keyframes* $K_{\text{ref}} \in K_1$ de los *keyframes* que comparten la mayoría de los puntos con la imagen actual.

Cada punto del mapa visto en K_1 y K_2 se busca en la imagen actual de la siguiente forma:

1. Proyectar el punto del mapa en la imagen actual, $\boldsymbol{x}$. Descartar si está fuera de los límites.
2. Calcular el ángulo entre el rayo de visión actual $\boldsymbol{v}$ y la dirección de visión media del punto del mapa $\boldsymbol{n}$. Descartar si $\boldsymbol{v} \cdot \boldsymbol{n} \leq \cos(60^{\text{o}})$.
3. Calcular la distancia d del punto del mapa al centro de la cámara. Descartar si está fuera de la región de invariabilidad de escala $d \notin [d_{\min}, d_{\max}]$.
4. Calcular la escala en la imagen actual con el ratio $\frac{d}{d_{\min}}$.
5. Crear una ventana de búsqueda cuyo centro está en $\boldsymbol{x}$. Comparar el descriptor $\boldsymbol{D}$ del punto del mapa con los puntos de interés detectados pero no emparejados de la imagen actual, en la escala predicha, dentro de la ventana de búsqueda y emparejar el punto del mapa con el punto de interés si es el único que hay dentro de la ventana o si es el más similar (*test de ratio*) y si la distancia entre descriptores es menor de un umbral establecido.

Una vez se hayan emparejado los puntos del mapa, la posición y la orientación son optimizados con los puntos del mapa encontrados.

Por último, se decide si insertar un nuevo *keyframe* o no. Para insertar un nuevo *keyframe* se tienen que cumplir las siguientes condiciones:

- Se han procesado más de 20 imágenes desde la última relocalización.
- Se han procesado más de 20 imágenes desde que se insertó el último *keyframe*.
- En la imagen actual se han seguido al menos 50 puntos.
- En la imagen actual se han seguido menos del 90 % de puntos que en K_{ref}.

Cuando se inserta un nuevo *keyframe*, se actualiza el grafo de covisibilidad añadiendo un nuevo nodo y actualizando las aristas resultantes de los puntos compartidos con otros *keyframes*.

Una vez insertado el *keyframe*, se eliminan los puntos del mapa que no son robustos y se crean nuevos puntos del mapa triangulado los puntos ORB de los *keyframe* conectados en el grafo de covisibilidad, utilizando la bolsa de palabras para acelerar la búsqueda (sección 3.4.2) y descartando aquellos emparejamientos que no cumplan la restricción de la linea epipolar. Los puntos emparejados son triangulados y se añaden al mapa comprobando la profundidad positiva en ambas cámaras, el paralaje, el error de reproyección y la consistencia de la escala. Tras esto se realiza un *Bundle Adjustment* local (sección 3.4.5). Además, para mantener una reconstrucción compacta, se eliminan *keyframes* redundantes.

Durante el seguimiento de estos puntos es importante tener en cuenta los retos que implica trabajar con secuencias grabadas por un endoscopio. Uno de ellos es que se está trabajando muy cerca de las paredes de los órganos por lo que el campo de visión es menor, así que es natural que se detecten menos puntos en las imágenes. Debido a esto se han modificado los diferentes umbrales de puntos de ORB-SLAM2 para que trabaje con menos puntos. Puesto que el sistema trabaja con menos puntos, es necesario que estos sean más robustos. Por eso es importante conseguir emparejamientos fuertes. Estos emparejamientos se pueden conseguir cambiando los detectores y los descriptores (como veremos en la sección 4.2.3).

3.4.2. Bolsa de palabras y vocabulario inicial

Como ya se ha comentado en la sección 2.2, ORB-SLAM2 utiliza DBoW2 para crear el vocabulario y realizar la relocalización y los cierres de bucle.

El vocabulario que utiliza ORB-SLAM2 está creado a partir de los puntos ORB obtenidos de miles de imágenes genéricas. En este trabajo se van a evaluar diferentes combinaciones de detectores y descriptores por lo que es necesario crear nuevos vocabularios.

Para crear estos vocabularios, se necesita una base de datos de imágenes. En este trabajo se ha elegido la segunda versión del dataset *Kvasir* [33] que contiene 8.000 imágenes pertenecientes a pro-

cedimientos quirúrgicos. Se ha decidido elegir este *dataset* ya que contiene imágenes del interior del cuerpo humano. En la figura 3.19 se puede ver un ejemplo de las imágenes que contiene.

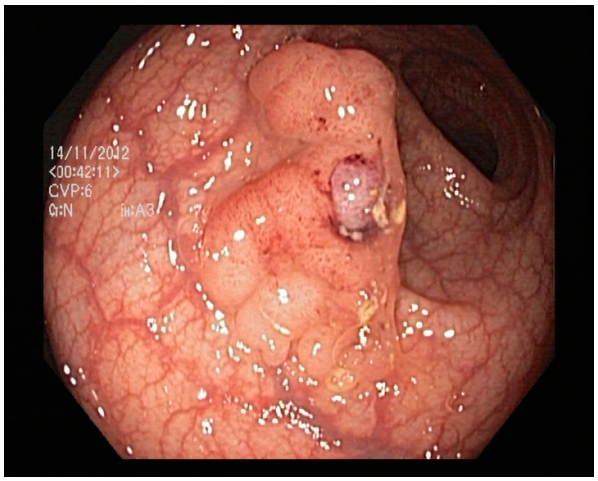


Figura 3.19: Imagen de ejemplo del *dataset Kvasir*.

La bolsa de palabras es una técnica que utiliza un vocabulario visual para convertir una imagen en un vector numérico disperso, permitiéndonos manejar grandes conjuntos de imágenes. El vocabulario visual se crea fuera de línea (de forma *offline*) discretizando el espacio descriptivo en W palabras visuales. En el caso de la bolsa jerárquica de palabras, el vocabulario se estructura como un árbol. Para construirlo, se extrae un gran conjunto puntos característicos de las imágenes de entrenamiento, independientes a las que se procesan en línea (*online*) más tarde. Los descriptores extraídos se discretizan primero en k_w *clusters* realizando agrupaciones de *k-medianas* (*k-medians*) con la semilla de *k-means++* [34]. Las medianas que resultan en un valor no binario se truncan a 0. Estos *clusters* forman el primer nivel de nodos en el árbol de vocabulario. Los niveles subsiguientes se crean repitiendo esta operación con los descriptores asociados a cada nodo, hasta L_w veces. Finalmente, obtenemos un árbol con W hojas, que son las palabras del vocabulario. A cada palabra se le da un peso según su relevancia en conjunto de imágenes de entrenamiento, disminuyendo el peso de aquellas palabras que son muy frecuentes y, por tanto, menos discriminatorias. Para ello, utilizamos *el término frecuencia de documento inverso a la frecuencia* (*tf-idf*) [35]. Para convertir una imagen I_t, tomada en el tiempo t, en un vector de bolsa de palabras $\mathbf{v}_t \in \mathbb{R}^W$, los descriptores de los puntos característicos detectados recorren el árbol desde la raíz hasta las hojas, seleccionando en cada nivel los nodos intermedios que minimizan la distancia Hamming.

Para calcular la similitud entre dos vectores de la bolsa de palabras $\mathbf{v}_1$ y $\mathbf{v}_2$, se calcula la puntuación

$L1$ (*L1-score*) $s(\mathbf{v}_1, \mathbf{v}_2)$, cuyo valor está en $[0, 1]$:

$$s(\mathbf{v}_1, \mathbf{v}_2) = 1 - \frac{1}{2}\left|\frac{\mathbf{v}_1}{|\mathbf{v}_1|} - \frac{\mathbf{v}_2}{|\mathbf{v}_2|}\right| \tag{3.21}$$

Junto con la bolsa de palabras, se mantiene un índice invertido. Esta estructura almacena para cada palabra del vocabulario una lista de imágenes donde está presente. Esto es muy útil cuando se consulta la base de datos, ya que permite realizar comparaciones sólo con aquellas imágenes que tienen alguna palabra en común con la imagen de la consulta. El índice invertido se actualiza cuando una nueva imagen se añade a la base de datos.

Además, también se hace uso de un índice directo para almacenar los puntos característicos de cada imagen. Se separan los nodos del vocabulario según su nivel l en el árbol, empezando por las hojas, con nivel $l = 0$, y terminando en la raíz, $l = L_w$. Para cada imagen I_t, se almacena en el índice directo los nodos del nivel l que son antepasados (nivel mayor, están más cerca de la raíz) de las palabras presentes en I_t, así como la lista de puntos característicos f_{tj} asociadas a cada nodo. ORB-SLAM2 utiliza el índice directo para acelerar el cálculo de las correspondencias entre dos conjuntos de puntos característicos ORB, se puede limitar los emparejamientos de fuerza bruta a sólo aquellos puntos característicos que pertenecen al mismo nodo del árbol de vocabulario en un determinado nivel. Se utiliza este truco a la hora de buscar emparejamientos para triangular nuevos puntos, y en la detección de bucles y relocalización.

La estructura del vocabulario con el árbol, el índice inverso y el índice directo se puede ver en la figura 3.20.

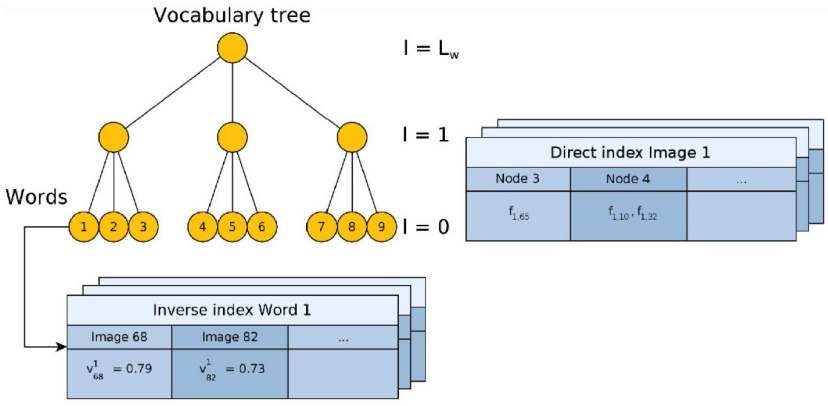


Figura 3.20: Estructura del vocabulario DBoW2 [36].

3.4.3. Relocalización

Si se pierde el seguimiento, la imagen es convertida a bolsa de palabras y se realiza una consulta en la base de datos usando el índice invertido para obtener los candidatos para la relocalización global.

Después se realizan iteraciones de RANSAC para cada *keyframe* posible y se intenta encontrar la posición y orientación de la cámara utilizando el algoritmo PnP [37]. Si se encuentra una posición y orientación con suficientes *inliers*, se optimiza la posición y orientación y se realiza una búsqueda guiada para conseguir más emparejamientos con los puntos del mapa del *keyframe* candidato. Por último, se vuelve a optimizar la posición y orientación y, si es soportada por suficientes *inliers*, el seguimiento continúa y el estado del sistema pasa a ser "*trakcing*" En caso contrario el estado del sistema sigue siendo "perdido" y se realiza relocalización en cada nueva imagen.

3.4.4. Cierre de bucle

Las trayectorias estimadas por sistemas SLAM monoculares suelen tener bastante deriva tanto por imprecisión como por pérdida de escala. Por ello es necesario detectar y cerrar bucles, lo que permite mejorar las trayectorias estimadas. Esta es la diferencia entre SLAM y odometría visual. Un cierre de bucle implica una optimización del mapa y de las posiciones y orientaciones estimadas de la cámara a lo largo de una trayectoria cuando el sistema detecta que está explorando una zona que ya ha sido visitada anteriormente en la trayectoria (reconocimiento de lugar o *place recognition*), sección 2.2.

En ORB-SLAM2, cada vez que se inserta un nuevo *keyframe*, se compara, en un hilo de ejecución en paralelo, si éste es similar a alguno de los *keyframes* almacenadas en la base de datos. Para ello se utiliza DBoW2, para así convertir la imagen a una bolsa de palabras y de esta forma realizar consultas a la base de datos.

El primer paso es detectar si hay un bucle. Para ello se buscan candidatos en la base de datos utilizando el último *keyframe* procesado, sus vecinos y el grafo de covisibilidad. Para aceptar un candidato, se deben detectar tres candidatos consecutivos consistentes (conectados en el grafo de covisibilidad).

Los pasos seguidos por ORB-SLAM2 para cerrar el bucle se pueden ver en la figura 3.21. Si se detecta un bucle (paso 1), se calcula la transformación de similitud (paso 2) que informa sobre la deriva acumulada en el bucle y se alinean ambos lados del bucle. Para ello se calculan las correspondencias entre los descriptores ORB asociados a los puntos del mapa del *keyframe* actual (K_i) y los asociados a los *keyframes* candidados (K_l). En este punto hay correspondencias entre puntos 3D por cada candidato. Se realizan iteraciones RANSAC con cada candidato tratando de encontrar la transformación de similitud usando el método de Horn [38]. Si se encuentra una similitud $\boldsymbol{S}_{il}$ con suficientes *inliers*, se optimiza y se realiza una búsqueda guiada para encontrar más correspondencias. Se vuelve a optimizar y si $\boldsymbol{S}_{il}$ es apoyada por suficientes *inliers*, se acepta el candidato K_l. Con la transformación de similitud $\boldsymbol{S}_{il}$ se corrige la posición y orientación del *keyframe* actual ($\boldsymbol{T}_{iw}$) y esta corrección se propaga a todos los vecinos de éste, concatenando las transformaciones de forma que ambos extremos del bucle quedan alineados (paso 3). Después, los puntos del mapa duplicados se fusionan (paso 4) y se actualiza el grafo

de covisibilidad. Por último, se realiza una optimización del grafo para conseguir consistencia global (paso 5). Esta optimización se realiza sobre un grafo esencial, es decir, un sub-grafo más pequeño del grafo de covisibilidad en el que se conservan sólo las aristas más fuertes (con un mayor número de puntos comunes). Esta optimización distribuye el error del cierre de bucle sobre todo el grafo.

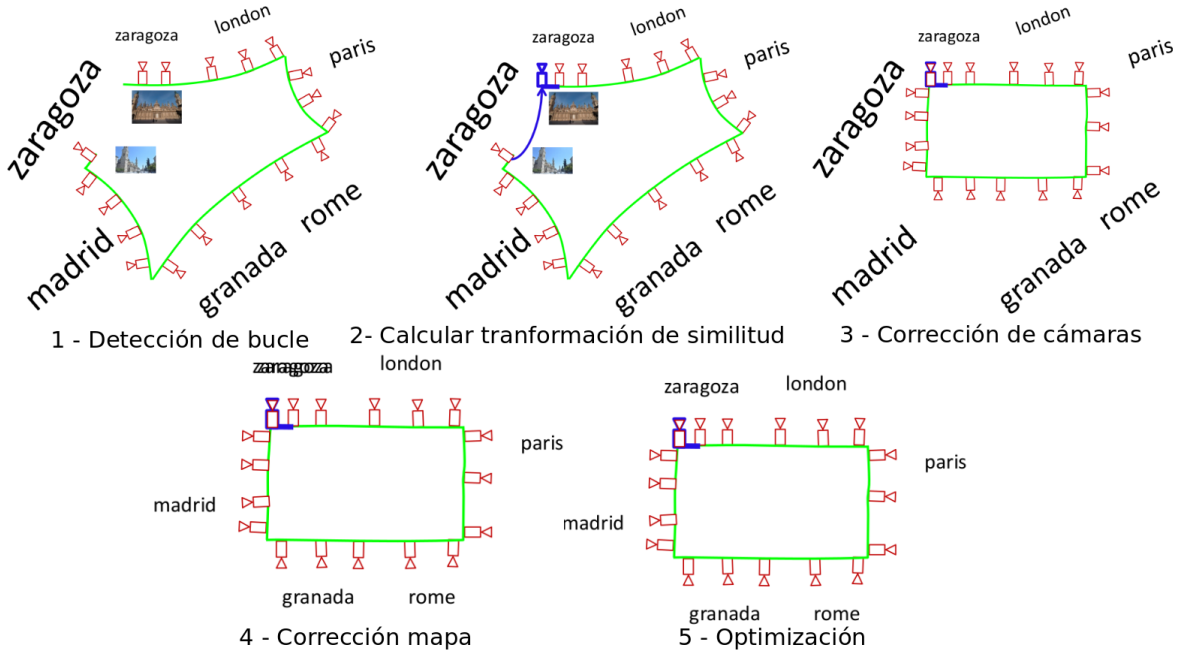


Figura 3.21: Pasos seguidos en el cierre de bucle [11].

Aunque tanto en la relocalización como en el cierre de bucle se utiliza la bolsa de palabras para buscar *keyframes* candidatos, estos procesos son muy diferentes. A diferencia de la relocalización que sólo se realiza a frecuencia de imagen cuando el sistema está perdido, el cierre de bucle se realiza a frecuencia de *keyframe* y sólo cuando se ha detectado que se han obtenido candidatos conectados en el grafo de covisibilidad durante tres *keyframes* seguidos. De esta forma ORB-SLAM2 asegura que el cierre de bucle sólo es ejecutado cuando realmente hay uno, evitando falsos positivos. Además, como ya se ha explicado las optimizaciones que conllevan cada uno de estos procesos son muy diferentes, siendo la relocalización más sencilla que el cierre de bucle.

3.4.5. Optimización

ORB-SLAM2 realiza *Bundle Adjustment (BA)* para optimizar tanto los puntos en el espacio 3D $\boldsymbol{X_{w,j}} \in \mathbb{R}^3$ como las posiciones y orientaciones de la cámara $\boldsymbol{T_{iw}} = [\boldsymbol{R}_{iw}\ \boldsymbol{t}_{iw}] \in SE(3)$ donde w

[11] Imágenes obtenidas de `https://moodle.unizar.es/add/pluginfile.php/2089614/mod_resource/content/4/6.%20SLAM_Visual.pdf`

representa el sistema de referencia global. Para ello se minimiza el error de reproyección con respecto a los puntos de interés emparejados $\boldsymbol{x}_{i,j} \in \mathbb{R}^2$. El error de la observación de un punto del mapa j en el *keyframe* i es:

$$\boldsymbol{e}_{i,j} = \boldsymbol{x}_{i,j} - \boldsymbol{\pi}_i(\boldsymbol{T_{iw}}, \boldsymbol{X_{w,j}}) \tag{3.22}$$

donde $\boldsymbol{\pi}_i$ es la función de proyección:

$$\boldsymbol{\pi}_i(\boldsymbol{T_{iw}}, \boldsymbol{X_{w,j}}) = \begin{bmatrix} f_{i,x}\frac{x_{i,j}}{z_{i,j}} + p_x \\ f_{i,y}\frac{y_{i,j}}{z_{i,j}} + p_y \end{bmatrix} \tag{3.23}$$

$$\begin{bmatrix} x_{i,j} \\ y_{i,j} \\ z_{i,j} \end{bmatrix} = \boldsymbol{R}_{i,w}\boldsymbol{X}_{w,j} + \boldsymbol{t}_{iw} \tag{3.24}$$

donde $\boldsymbol{R}_{i,w} \in SO(3)$ y $\boldsymbol{t}_{iw} \in \mathbb{R}^3$ son respectivamente las partes de rotación y traslación de la matriz $\boldsymbol{T_{iw}}$ y $(f_{i,x}, f_{i,y})$ y (p_x, p_y) son la distancia focal y el punto principal asociado a la cámara. La función de coste a minimizar es:

$$\boldsymbol{C} = \sum_{i,j} \rho_h(\boldsymbol{e}_{i,j}^T \boldsymbol{\Omega}_{i,j}^{-1} \boldsymbol{e}_{i,j}) \tag{3.25}$$

donde ρ_h es la función de coste de Huber y $\boldsymbol{\Omega}_{i,j} = \sigma_{i,j}^2 \boldsymbol{I}_{2\times 2}$ es la matriz de covarianza asociada a la escala en el que el punto fue detectado.

En caso de *BA* completo (usado en la inicialización del mapa), se optimizan las posiciones y orientaciones todos los *keyframes* y puntos del mapa a excepción del primer *keyframe* que queda fijado como el origen. En caso de *BA* local, todos los puntos que estén en el área local son optimizados, mientras que un subconjunto de *keyframes* es fijo. En la optimización de la posición, todos los puntos son fijados y sólo se optimizan las posiciones y orientaciones de la cámara.

3.4.6. Inicialización

Como ya se ha comentado, la baja repetibilidad de FAST puede causar malos emparejamientos. Estos malos emparejamientos, junto con la dificultad de encontrar buenos puntos, pueden llevar a una mala inicialización, creando un mapa inicial poco preciso que llevará a estimar la posición y orientación de forma incorrecta y a *trackear* o seguir unos puntos mal triangulados. Para ello, además de probar los diferentes detectores y descriptores mencionados, en este trabajo se han realizado varias modificaciones en la inicialización para intentar estimar un mapa inicial más robusto.

La inicialización de ORB-SLAM2 sigue los siguientes pasos:

1. Detectar puntos en la imagen actual y emparejarlos con los puntos detectados en la imagen de referencia. Si no se han encontrado suficientes emparejamientos, actualizar la imagen de referencia por la imagen actual y reiniciar la inicialización.

2. Calcular en hilos de ejecución paralelos la matriz fundamental y la homografía y calcular una puntuación para cada modelo (S_F, matriz fundamental y S_H, homografía). Si no se consiguen *inliers* suficientes, reiniciar inicialización.

3. Seleccionar el modelo que más se ajuste. Si la escena es plana, casi plana o hay bajo paralaje, se puede explicar por una homografía. Una escena no plana con suficiente paralaje sólo puede ser explicada por la matriz fundamental, aunque si un subconjunto de emparejamientos se encuentran en un plano o tienen bajo paralaje, éstos pueden ser explicados mediante una homografía. Para decidir qué modelo usar, ORB-SLAM2 utiliza la siguiente ecuación:

$$R_H = \frac{S_H}{S_H + S_F} \tag{3.26}$$

   y selecciona la homografía si $R_H > 0{,}4$, en caso contrario selecciona la matriz fundamental.

4. Una vez selecionado el modelo, se triangulan los puntos de acuerdo con ese modelo.

5. Finalmente se realiza *Bundle Adjustment* para refinar la reconstrucción inicial.

De este proceso hay dos puntos que se han modificado: el paralaje mínimo necesario para la inicialización y la heurística. Se ha aumentado el paralaje mínimo, de esta forma se triangularán los puntos del mapa inicial de forma más robusta y precisa. Además, se ha modificado la heurística mencionada ya que los órganos no son estructuras planas. Debido a esto sólo se seleccionará la homografía si $R_H > 0{,}6$, aumentando la probabilidad de que se inicialice con la matriz fundamental pero dejando la posibilidad de que se inicialice con la homografía en caso de que un subconjunto de emparejamientos pertenezca a un plano.

3.5. Resumen de modificaciones

El principal problema que tiene ORB-SLAM2 para estimar el mapa y la posición del endoscopio es la dificultad de encontrar puntos FAST en el interior del cuerpo. Esto es debido a la ausencia de esquinas en éste. Por este motivo en este trabajo se ha probado la combinación de los detectores FAST, A-KAZE y Shi-Tomasi con los descriptores BRIEF y A-KAZE.

Al probar diferentes combinaciones de detectores y descriptores es necesario cambiar el vocabulario de ORB-SLAM2 para cada una de las combinaciones. Para crear estos vocabularios se ha utilizado el *dataset Kvasir* que contiene imágenes de diferentes procedimientos quirúrgicos. Para la creación de estos vocabularios se ha utilizado DBoW2.

Además, otro problema que se encuentra en las secuencias grabadas por un endoscopio son los cambios bruscos de iluminación pudiendo encontrar imágenes con mucha iluminación, imágenes oscuras,

imágenes con buena iluminación o imágenes con partes oscuras y partes iluminadas en la misma secuencia. Debido a esto en este trabajo se ha procesado la imagen antes de ser tratada por ORB-SLAM2. En este procesado se ha utilizado CLAHE para mejorar la iluminación de ésta de forma adaptativa y se ha seleccionado el canal verde de la imagen mejorada. De esta forma se consigue estimar un mayor porcentaje de la trayectoria ya que el sistema no se pierde al visitar zonas más oscuras del órgano. Además, para evitar detectar puntos en los reflejos y que estos se emparejen y triangulen de forma incorrecta, se ha creado una máscara que se utiliza para ignorar los puntos detectados en los reflejos.

En las secuencias grabadas por un endoscopio, éste trabaja muy cerca de las paredes del órgano que se está explorando por lo que el campo de visión es pequeño. Debido ha esto se han modificado los diferentes umbrales de puntos para que el sistema trabaje con un menor número de puntos. En concreto se ha reducido el número de *inliers* necesarios para seguir el mapa en un 60 % al igual que los *inliers* necesarios para realizar relocalización.

Por último, se ha modificado la inicialización. Dado que no se está trabajando con escenas planas, se ha modificado la heurística para que se seleccione la homografía sólo si R_H >0,6. Además, también se ha aumentado el paralaje mínimo para inicializar. Este aumento ha sido de 1º a 1,4º de esta forma se triangularán los puntos de forma algo más robusta.

Capítulo 4

Evaluación

En este apartado se van a evaluar las diferentes modificaciones realizadas en ORB-SLAM2 y cómo los diferentes detectores y descriptores afectan al funcionamiento de éste en secuencias correspondientes a procedimientos de cirugía mínimamente invasiva.

En este tipo de operaciones hay una fase inicial en la que se explora el órgano; después de esta fase el endoscopio se extrae y se inserta varias veces a lo largo de la operación. Para evaluar ORB-SLAM2 se ha usado una secuencia en la que se explora la parte inferior de la vejiga, se extrae el endoscopio y se vuelve a insertar para continuar explorando la misma zona. De esta forma se puede evaluar el mapa creado, la trayectoria y la relocalización.

Además, se han obtenido diferentes *datasets* que se explicarán el los siguientes subapartados junto con los experimentos realizados y la forma de evaluar estos experimentos.

Los experimentos se han realizado en un ordenador portátil con 8 GB de memoria RAM y un procesador Intel(R) Core(TM) i7-5500U CPU@2.40 GHz.

4.1. Datasets

4.1.1. Dataset 1: Citoscopia

Este dataset corresponde a una secuencia grabada durante una citoscopia (eliminación de pólipos en la vejiga) en un paciente real. Esta operación es bastante limpia ya que, a diferencia de la operación de piedras en el riñón, en esta operación no se produce polvo que nuble la visión de la cámara. Además, en la figura 4.1 se puede observar cómo en estas imágenes hay zonas con características visuales (venas y pólipos) por lo que ORB-SLAM2 puede detectar puntos. Debido a esto, este ha sido el dataset utilizado para la evaluación.

Aunque no se tenga ni la calibración de este endoscopio ni el *ground truth*, esta secuencia se ha escogido por los siguientes motivos:

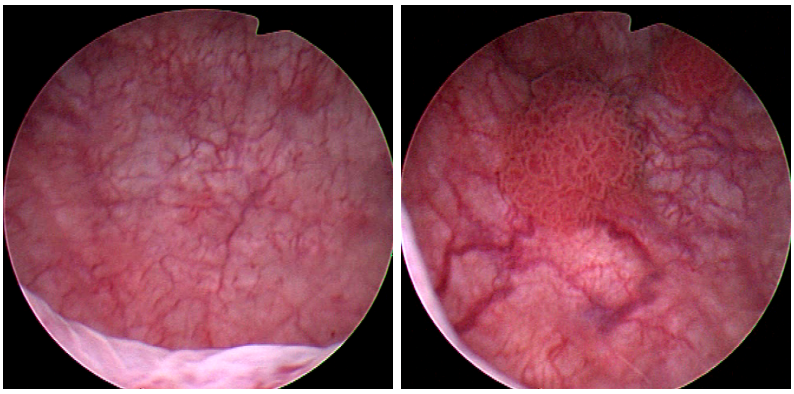


Figura 4.1: Imágenes tomadas por el endoscopio en el interior de una vejiga.

- Las imágenes de la secuencia no tienen mucho ruido.
- Es una secuencia que contiene tanto imágenes claras como imágenes oscuras perfectas para probar el pre-procesado del las imágenes.
- La secuencia contiene movimientos bruscos, reflejos y cambios de luz, algo que es muy habitual durante los procedimientos endoscópicos.
- En este secuencia se extrae y se re-inserta el endoscopio por lo que es ideal para probar la relocalización.

La secuencia seleccionada tiene un total de 600 imágenes. Estas imágenes se pueden dividir en diferentes categorías según su iluminación o lo que se visualiza en la imagen. Esta clasificación se puede ver en la tabla 4.1. El endoscopio está en el tubo que conecta el exterior del cuerpo del paciente con el interior de su vejiga (tubo utilizado para evitar heridas en los conductos y órganos que conectan el órgano a explorar con el exterior del cuerpo) durante 20 imágenes por lo que el mayor porcentaje de trayectoria estimada correctamente puede ser de un 96,7 %. También se ha de tener en cuenta que al re-insertar el endoscopio éste tarda unas imágenes en volver a la zona ya visitada, por lo que el sistema estará en el estado "perdido" durante más imágenes. Además, esta secuencia tiene 100 imágenes con baja iluminación. Uno de los principales retos es no perder el seguimiento de la cámara y de los puntos durante estas imágenes. Otro reto es no perder el seguimiento cuando entra en el campo de visión de la cámara la herramienta utilizada para eliminar los pólipos de la vejiga. El sistema de SLAM debe ser capaz de seguir estimando la trayectoria y el mapa 3D a pesar de que haya baja iluminación o de que la herramienta de eliminación de pólipos entre en el campo de visión. En caso de que se pierda, el sistema debe ser capaz de relocalizarse al volver a pasar por la misma zona o una zona anterior.

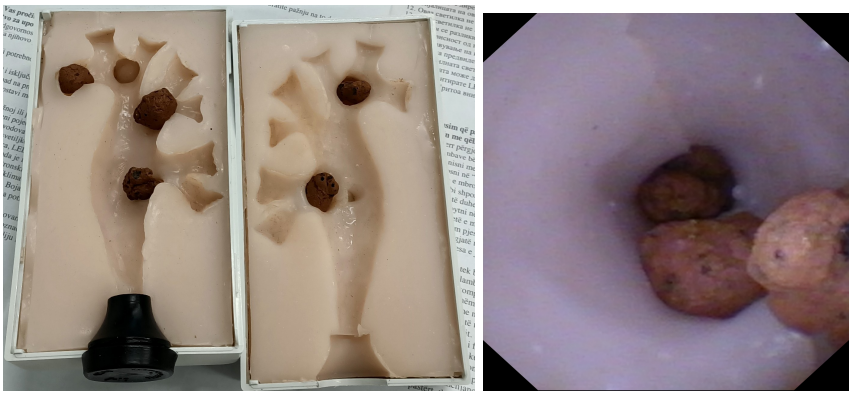
Observaciones	Número imágenes
Baja iluminación	100
Endoscopio en el tubo	20
Herramienta de eliminación de pólipos en el campo de visión	30
Imágenes con buena iluminación	450
Total	600

Cuadro 4.1: Número de imágenes de la secuencia.

4.1.2. Dataset 2: Ureteroscopia

Riñón sintético

Este trabajo de fin de máster ha sido realizado en el DLR (centro aeroespacial alemán). En las instalaciones de este centro hay un endoscopio flexible Olympus [12] y un par de riñones sintéticos que se encargaron al inicio de este proyecto. El problema encontrado en estos riñones es que aunque éstos tienen la misma estructura que un riñón real y tienen objetos que simulan las piedras que se han de quitar en la operación de extracción de piedras en el riñón, estos riñones *no son visualmente similares a un riñón real*. Los riñones sintéticos no tienen venas ni texturas (figuras 4.2(a) y 4.2(b)). Debido a esto, es imposible encontrar puntos característicos para emparejas y triangular, siendo casi imposible hacer funcionar ORB-SLAM2 en este dataset y siendo descartado por la falta de similitud con un dataset real. Nuestro objetivo como trabajo futuro es texturizar los riñones de silicona en la empresa productora Samed[13] o en la empresa especialista en silicona KauPo [14]. Una vez texturizados, planeamos usar estos riñones sintéticos para mostrar la interfaz al cirujano. Además, este *dataset* será importante por la alta precisión de la calibración de la cámara.



(a) Estructura del riñón sintético. (b) Interior del riñón sintético.

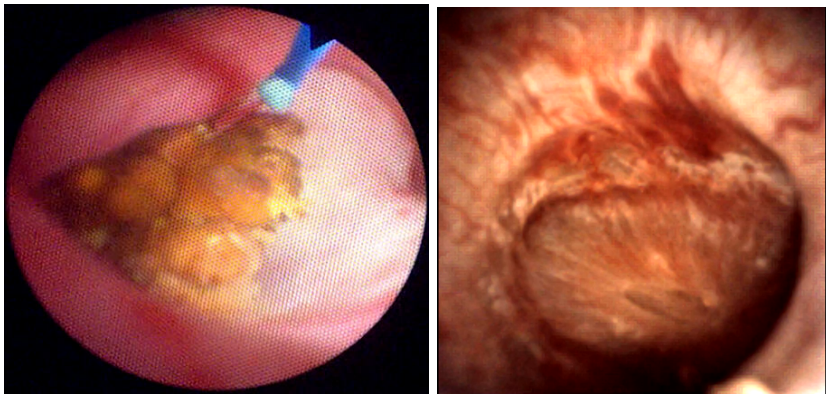
Figura 4.2: Imágenes riñón sintético.

[12] https://www.olympus-europa.com/medical/en/Products-and-Solutions/Products/Product/URF-V.html
[13] https://samed-dresden.de/endouro-trainer-ls50/
[14] https://www.kaupo.de/

Riñón real

Este dataset corresponde a una secuencia de una ureteroscopia (eliminación de piedras en el riñón) grabada con un endoscopio flexible en un paciente real. El problema de este dataset es que en las imágenes se pueden observar artefactos (figura 4.3(a)). Probablemente estos artefactos se hayan creado debido a la compresión del vídeo. En este caso ORB-SLAM2 detecta los puntos característicos en los artefactos siendo imposible inicializar. Debido a esto, este dataset también ha sido descartado. Como trabajo futuro vamos a obtener un vídeo equivalente sin artefactos para poder extender este sistema a diferentes procedimientos quirúrgicos.

También se consiguieron secuencias grabados con un endoscopio de un único uso en un paciente real. El problema con estos endoscopios es que la calidad de las imágenes es más baja. Además, estos vídeos siguen conteniendo artefactos (figura 4.3(b)) que impiden que se detecten los puntos correctos ya que se seguirán detectando puntos en estos artefactos. Debido a esto, este dataset también fue descartado.



(a) Endoscopio de varios usos. (b) Endoscopio de único uso.

Figura 4.3: Artefactos visibles en las imágenes correspondientes a secuencias de una ureteroscopia.

4.2. Experimentos

Para la evaluación de las modificaciones realizadas en ORB-SLAM2 y de los diferentes detectores y extractores se va a utilizar el dataset correspondiente a una secuencia de la operación de eliminación de pólipos en la vejiga. Debido a la falta de una trayectoria de validación precisa (*ground truth*), para medir la precisión del sistema, éste se ha evaluado en base a las siguientes medidas:

- Tiempo de ejecución medio por cada imagen.

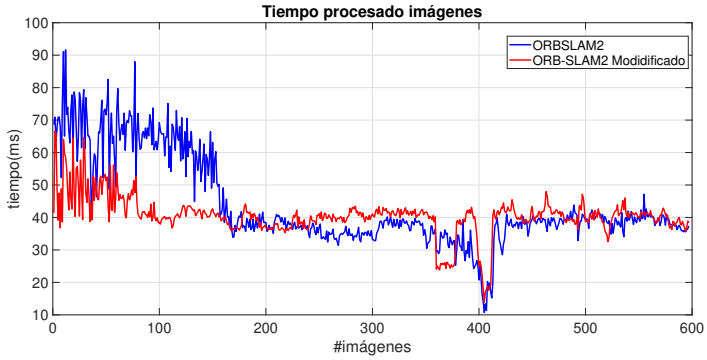
- Porcentaje medio de puntos del mapa seguidos en la nueva imagen con respecto al número de puntos detectados en ésta (emparejamiento activo para el seguimiento de los puntos del mapa).
- Número medio de imágenes necesarias para inicializar.
- Número medio de imágenes en las que el sistema está estimando la trayectoria y el mapa.
- Número medio de imágenes en las que el sistema está perdido.
- Capacidad del sistema para relocalizarse una vez el endoscopio es re-insertado o cuando se pierde debido a movimientos bruscos o cambios de iluminación.
- Mapa y trayectoria obtenidos.

En los experimentos realizados se ha ejecutado ORB-SLAM2 30 veces obteniendo la media del número de puntos, la media del número de imágenes en cada estado y el tiempo medio de procesado de cada imagen de estas ejecuciones. Además, para evaluar la relocalización se ha a evaluado la capacidad que tiene el sistema para relocalizar en la mejor y en la peor ejecución. La selección de la mejor y peor ejecución depende del número de imágenes en las que el sistema no está ni inicializando ni perdido. Este estado se ha llamado "Seguimiento" o "Tracking", durante este estado el sistema esta estimando tanto la posición y la orientación de la cámara como el mapa 3D a la vez que realiza el seguimiento del mapa local. La mejor ejecución corresponde a la ejecución en la que el sistema está en este estado durante más imágenes. La peor ejecución corresponde a la que está en este estado durante un menor número de imágenes.

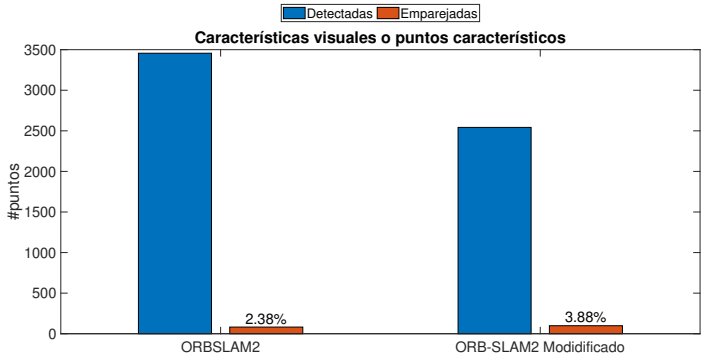
4.2.1. Modificaciones a ORB-SLAM2

Se han realizado varias modificaciones en ORB-SLAM2 (sección 3.5):

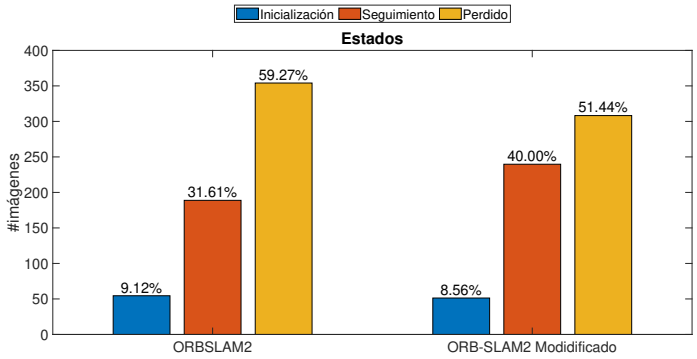
- Pre-procesado de las imágenes para reducir el ruido y aumentar el contraste. En ese pre-procesado se mejora la iluminación utilizando CLAHE y el canal verde de la imagen y creando una máscara para ignorar los puntos detectados en los reflejos.
- Modificación de los umbrales relacionados con el número de puntos (se trabaja con un número menor de características visuales debido a la dificultad de encontrar éstas en el interior de los órganos y al tener un campo de visión menor).
- Modificación del umbral de la heurística usada en la inicialización y del paralaje mínimo necesario para inicializar.



(a) Tiempo medio de procesado de imagen.



(b) Puntos característicos.



(c) Número medio de imágenes en cada estado.

Figura 4.4: Evaluación de las modificaciones realizadas en ORB-SLAM2.

En la figura 4.4(a) se puede observar que estas modificaciones apenas afectan al tiempo de procesado de cada imagen, el sistema sigue funcionando en tiempo real, el tiempo de cómputo es de unos 40 ms (25 fps, *frames per second*, imágenes por segundo). Sin embargo, en la figura 4.4(b) se puede observar que

aunque se detecten menos puntos característicos, la cantidad de puntos del mapa seguidos aumenta, pasando de 2,4 % de 3500 a un 3,9 % de 2500 (de 84 a 97), es decir, se consiguen más puntos del mapa a partir de menos puntos detectados en la imagen por lo que *los puntos detectados son más robustos*.

Por último, en la figura 4.4(c) se puede observar que con las modificaciones realizadas se consigue realizar el seguimiento y la estimación del mapa durante un 10 % más de imágenes debido a que, como los puntos detectados son más robustos, los emparejamientos son más fuertes por lo que la estimación del mapa y de la posición y orientación de la cámara es más precisa. Además, con las modificaciones realizadas, es más difícil que el sistema se pierda debido a cambios de iluminación (por el uso de CLAHE para aumentar el contraste).

A pesar de que el sistema mejora con estas modificaciones, sólo se estima un 40 % de la trayectoria. Debido a esto, se va a evaluar el uso de otros detectores y descriptores partiendo del sistema de ORB-SLAM2 modificado.

4.2.2. Vocabulario

ORB-SLAM2 utiliza DBoW2 para crear el vocabulario necesario para la relocalización y los cierres de bucle. ORB-SLAM2 trabaja con puntos FAST-ORB por lo que el vocabulario proporcionado es válido cuando se trabaja con este tipo de puntos. Este vocabulario está creado a partir de miles de imágenes genéricas. Para evaluar el sistema con las diferentes combinaciones de detectores y descriptores es necesario crear un nuevo vocabulario para cada una de ellas.

El nuevo vocabulario ha sido creado usando una base de datos de imágenes médicas y el software DBoW2. En este apartado se va a evaluar cómo afecta el cambio de vocabulario al funcionamiento del sistema. En la figura 4.5 se puede ver el número medio y porcentaje de imágenes en el que el sistema está en cada estado. Se puede observar que con el vocabulario genérico el sistema consigue estimar en torno a un 55 % de la trayectoria mientras que con el vocabulario médico se estima un porcentaje menor. Esto es debido a que el vocabulario también se utiliza para emparejar puntos ORB a la hora de triangular nuevos puntos para añadirlos al mapa (sección 3.4.2).

Además, en las figuras 4.6(a) y 4.6(b) se puede observar el estado en el que el sistema está en cada imagen en la mejor de las ejecuciones. En estas figuras se puede observar que la relocalización tiene una pequeña mejora cuando se usa el vocabulario genérico. También se puede observar que necesita más imágenes para inicializar, esto puede ser debido a que el sistema consigue inicializar con dos imágenes pero que no se consiguen triangular puntos suficientes nuevos con el árbol del vocabulario médico y por lo tanto se descarte la inicialización.

Para el vocabulario lo ideal sería conseguir una base de datos de imágenes de diferentes endoscopias de mayor tamaño que la base de datos de *Kvasir*. A falta de una base de datos más extensa, los vocabularios se han creado con las imágenes de *Kvasir*.

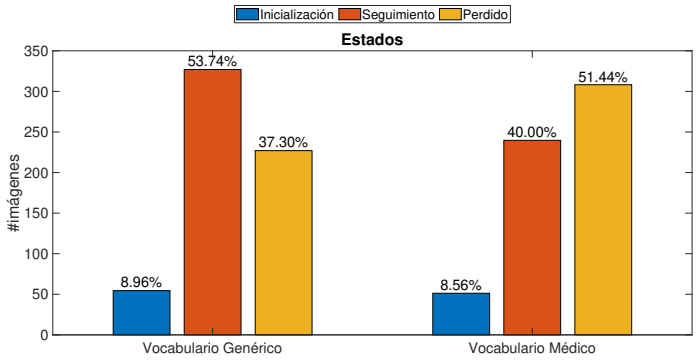
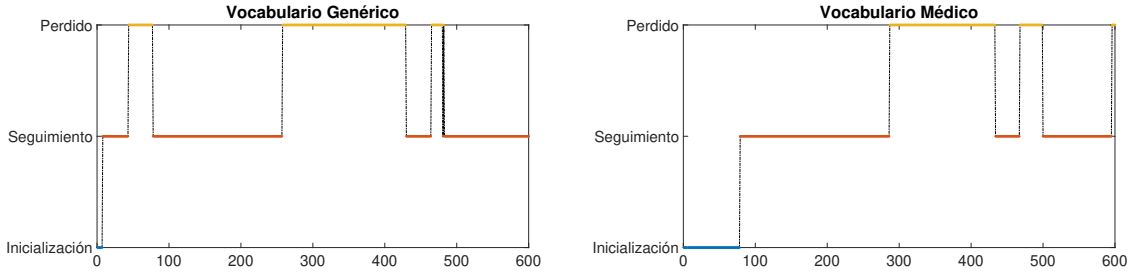


Figura 4.5: Número medio de imágenes en cada estado con cada vocabulario.



(a) Vocabulario ORB obtenido a partir de imágenes genéricas.

(b) Vocabulario ORB obtenido a partir de imágenes médicas.

Figura 4.6: Evaluación de la relocalización según el vocabulario.

4.2.3. Detectores y descriptores

A pesar de las modificaciones realizadas en ORB-SLAM2, el sistema sigue sin ser capaz de estimar más del 50 % del mapa y de la trayectoria. Esto puede ser debido a las desventajas de utilizar los puntos característicos FAST. Aunque estos sean rápidos de calcular, tienen baja repetibilidad, impidiendo realizar emparejamientos fuertes o realizar el seguimiento de estos a lo largo de varias imágenes. Por este motivo se han analizado las combinaciones de los diferentes detectores y descriptores mencionados en la sección 3.2.1. En este apartado se van a evaluar las ventajas y desventajas de utilizar cada uno de ellos utilizando la notación detector-descriptor a la hora de referirse a cada una de las combinaciones. En caso de que no se use ni el descriptor ni el detector A-KAZE pero se use el espacio de escala no lineal, esté vendrá indicado entre paréntesis junto con el nombre detector-descriptor (NLS).

En la figura 4.7 se puede observar el tiempo de procesado de las imágenes dependiendo de la combinación de detector-descriptor utilizada. En esta figura se puede observar que A-KAZE-A-KAZE es el más caro de calcular y FAST-ORB es el más barato. Si se utiliza FAST-ORB con el espacio de

escala no lineal de A-KAZE, es más rápido porque este es más pequeño que la representación piramidal original. Además, también se puede observar que el descriptor A-KAZE es más caro de calcular y emparejar que el descriptor ORB. Esta diferencia se puede observar al comparar, por ejemplo, A-KAZE-A-KAZE con A-KAZE-ORB, hay una diferencia de 100 ms.

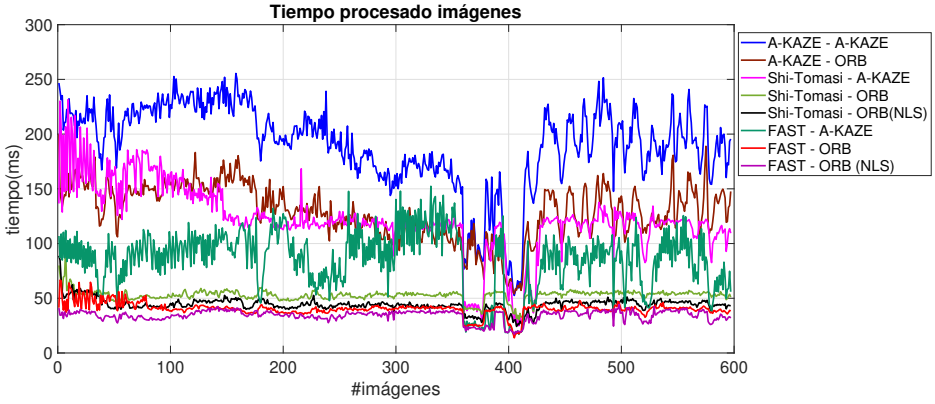


Figura 4.7: Tiempo de procesado de imagen según la combinación detector-descriptor.

En la figura 4.8 se puede observar que con A-KAZE-A-KAZE y A-KAZE-ORB se consiguen emparejar un mayor número de puntos del mapa, aproximadamente unos 300 puntos que si se compara con FAST-ORB, éste sólo consigue emparejar 97 puntos. Esto indica que el detector A-KAZE detecta puntos más robustos que los detectores FAST o Shi-Tomasi, aunque el detector Shi-Tomasi es más robusto que FAST.

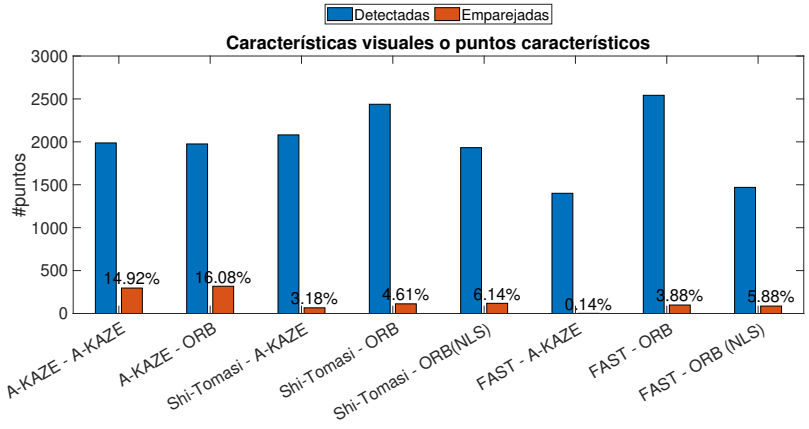


Figura 4.8: Porcentaje de puntos del mapa emparejados con los puntos detectados en la imagen según la combinación detector-descriptor.

Además, en la figura 4.9 se puede ver que tanto con A-KAZE-A-KAZE como con A-KAZE-ORB

se consigue estimar cerca del 80 % de la trayectoria. Aunque con A-KAZE-ORB se consiga estimar un 10 % menos de la trayectoria, éste tiene un tiempo de computación (150 ms) mucho más pequeño que A-KAZE-A-KAZE (250 ms), siendo la mejor combinación en relación al porcentaje de la trayectoria estimada y tiempo.

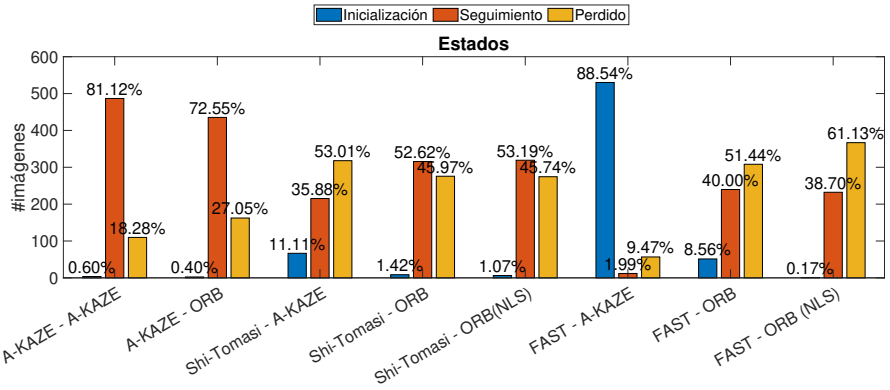
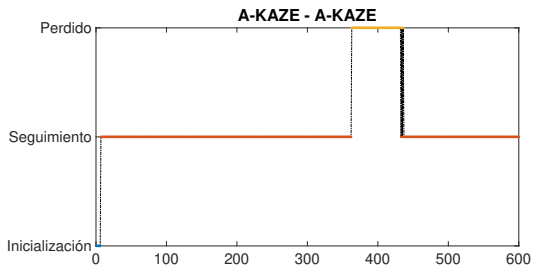


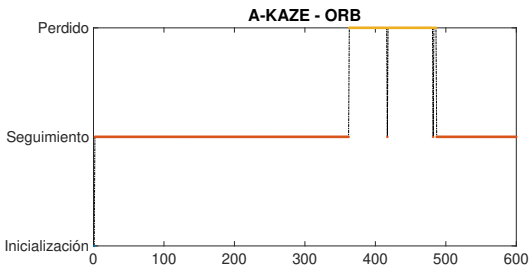
Figura 4.9: Número de imágenes en cada estado según la combinación detector-descriptor.

Un punto importante a evaluar con estas combinaciones es la relocalización ya que si ORB-SLAM2 estima de manera precisa el mapa y la trayectoria pero no realiza relocalización, no se consideraría un buen sistema SLAM. Para evaluar la relocalización en la figura 4.10 se pueden ver los estados en el que el sistema se encuentra en cada imagen de la secuencia médica en la mejor ejecución de ORB-SLAM2. En esta figura se puede observar que A-KAZE-A-KAZE, A-KAZE-ORB y Shi-Tomasi-ORB son capaces de realizar relocalización de forma casi perfecta. Obteniendo los mejores resultados con A-KAZE-A-KAZE. Mientras que FAST-ORB tiene una ejecución bastante similar y precisa aunque se pierde durante unas imágenes tras la relocalización. Pero estas gráficas corresponden a las mejores ejecuciones (ejecuciones en las que se ha estimado la mayor parte de la trayectoria). En la figura 4.11 se puede observar la relocalización correspondiente a las peores ejecuciones (ejecuciones en las que se ha estimado la menor parte de la trayectoria). En esta figura se puede observar que, aunque al final de la secuencia A-KAZE-A-KAZE y A-KAZE-ORB se pierden, ambos sigue siendo capaz de realizar relocalización cuando el endoscopio es re-insertado y son capaces de estimar una gran parte de la trayectoria. Sin embargo, FAST-ORB es incapaz de realizar relocalización. Esto puede ser debido a la baja repetibilidad de FAST por la que se realizan emparejamientos de puntos que no corresponden con lo que se introducen en el mapa puntos mal triangulados y, por lo tanto, se estima la posición y la orientación del endoscopio de forma errónea. Shi-Tomasi-ORB también se relocaliza pero se acaban perdiendo y es sólo es capaz de estimar un pequeño porcentaje de la trayectoria.

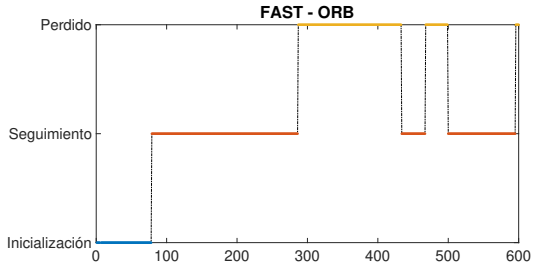
Además, en la figura 4.12, se pueden observar los diferentes mapas estimados. Debido a la falta de



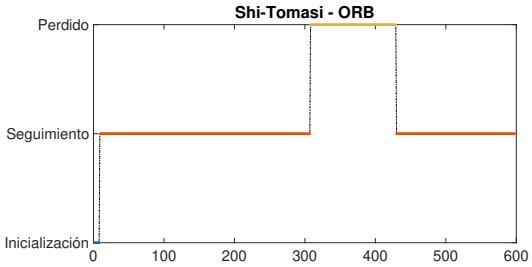
(a) A-KAZE - A-KAZE.



(b) A-KAZE - ORB.

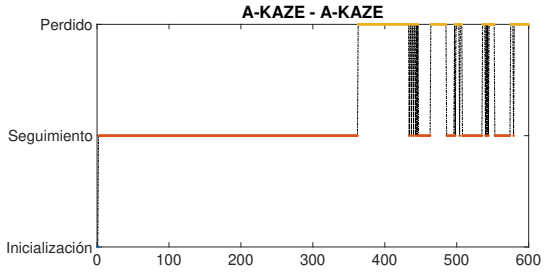


(c) FAST - ORB.

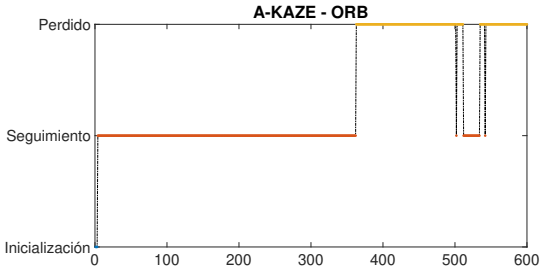


(d) Shi-Tomasi - ORB.

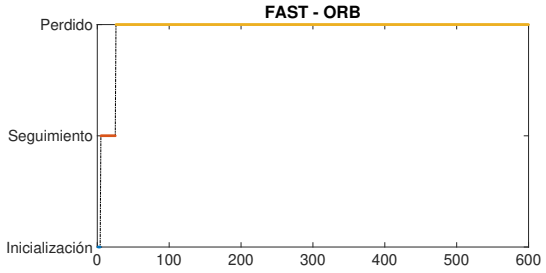
Figura 4.10: Evaluación de la relocalización con diferentes detectores y descriptores (mejor ejecución).



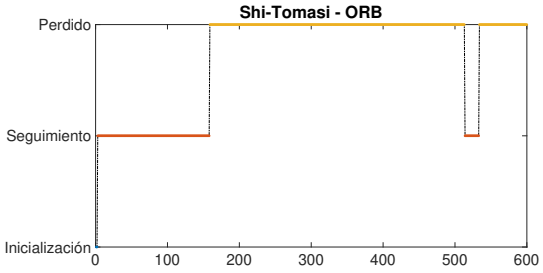
(a) A-KAZE - A-KAZE.



(b) A-KAZE - ORB.



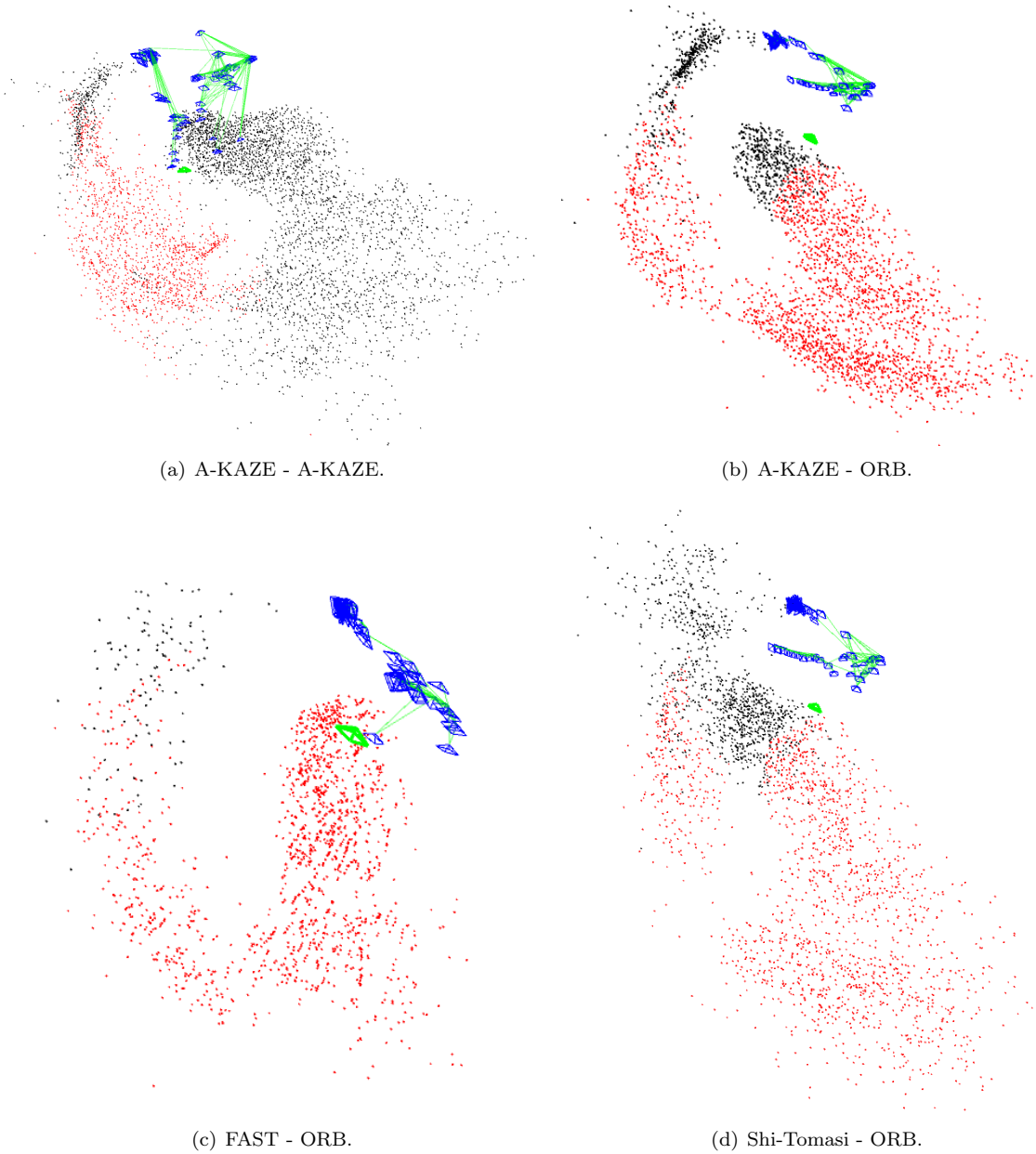
(c) FAST - ORB.



(d) Shi-Tomasi - ORB.

Figura 4.11: Evaluación de la relocalización con diferentes detectores y descriptores (peor ejecución).

ground truth no se puede decir si la estimación del mapa y la trayectoria son más o menos precisos, la única conclusión que se puede sacar de estos mapas es que parece que se asemejen a una superficie curva y que todos los puntos están en esa superficie.



(a) A-KAZE - A-KAZE.

(b) A-KAZE - ORB.

(c) FAST - ORB.

(d) Shi-Tomasi - ORB.

Figura 4.12: Mapas estimados con los diferentes detectores y descriptores.

Aunque A-KAZE-A-KAZE sea el más caro de calcular, es con el que mejores resultados se obtienen

ya que es capaz de estimar el mapa y la trayectoria de un mayor porcentaje de la secuencia y es capaz de realizar relocalización cuando el endoscopio es extraído y re-introducido. Sin embargo, con A-KAZE-ORB sólo se estima un 10 % menos de la trayectoria que A-KAZE-A-KAZE pero éste lo hace en mucho menos tiempo, también es capaz de realizar relocalización de forma precisa y el mapa estimado es bastante similar al estimado por A-KAZE-A-KAZE, siendo la mejor opción de combinación detector-descriptor.

En conclusión, cambiando el detector y el descriptor de ORB-SLAM2 se ha conseguido estimar cerca de un 75 % de la trayectoria sin aumentar demasiado el tiempo usando la combinación A-KAZE-ORB cuyo mapa estimado parece bastante preciso.

4.2.4. A-KAZE

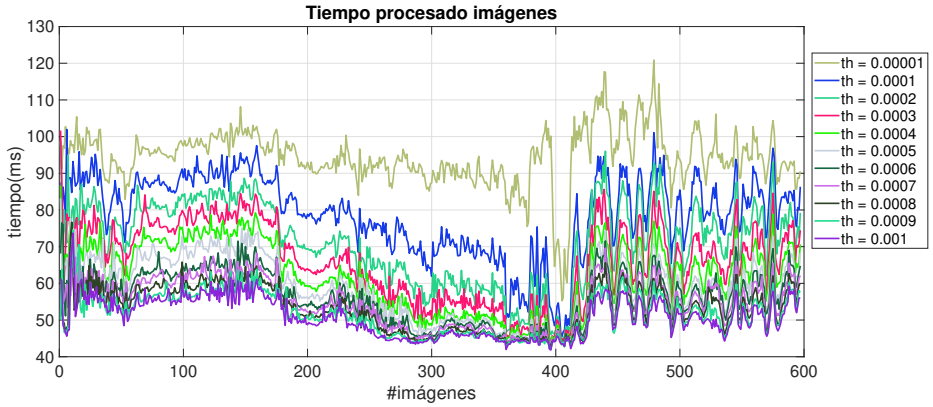
En las pruebas del apartado anterior se ha utilizado el detector y descriptor A-KAZE[15] con los parámetros por defecto, excepto el umbral para detectar si un punto es un punto característico que se ha disminuido para que el sistema pudiera funcionar. En este apartado se va a evaluar si es posible mejorar los resultados obtenidos con A-KAZE-ORB sin aumentar el tiempo. Para ello se van a evaluar los siguientes parámetros del detector A-KAZE:

- Umbral (*threshold*) que se utiliza en el detector A-KAZE para determinar si un punto es característico o no (valor por defecto 10^{-3}).
- Niveles: número de niveles del espacio de escala no lineal (disminución del tamaño de la imagen a la mitad, valor por defecto 8, aunque por el tamaño de la imagen se reduce a 3).
- Subniveles: número de subniveles del espacio de escala no lineal (difuminado de la imagen en cada nivel, valor por defecto 4).

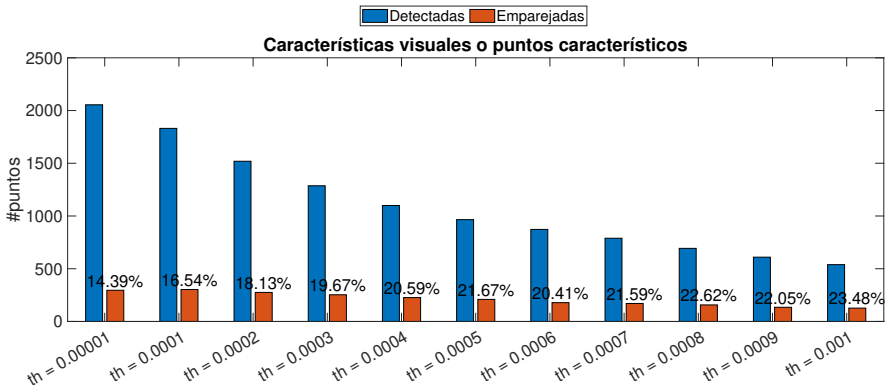
Además, para estos experimentos, se ha decidido añadir un filtrado de puntos en el que, a la hora detectar puntos de interés en el espacio de escala no lineal, se ignoran los dos primeros subniveles de la octava o nivel 0 de éste. De esta forma se asegura evitar detectar puntos en el ruido que contienen las imágenes tomadas por el endoscopio.

En la figura 4.13(a) se puede ver el tiempo medio que el sistema tarda en procesar una imagen. Lo primero que hay que destacar es que al utilizar el filtro de puntos mencionado se ha conseguido reducir el tiempo a unos 90 ms (sin el filtro son unos 150 ms). Además, se puede observar que conforme disminuye el umbral del detector A-KAZE, aumenta el tiempo. Esto es debido a que se detectan más puntos característicos por lo que se trabaja con un número mayor de puntos. Esto se puede observar en la figura 4.13(b). En esta figura también se puede ver que aunque se detecten menos

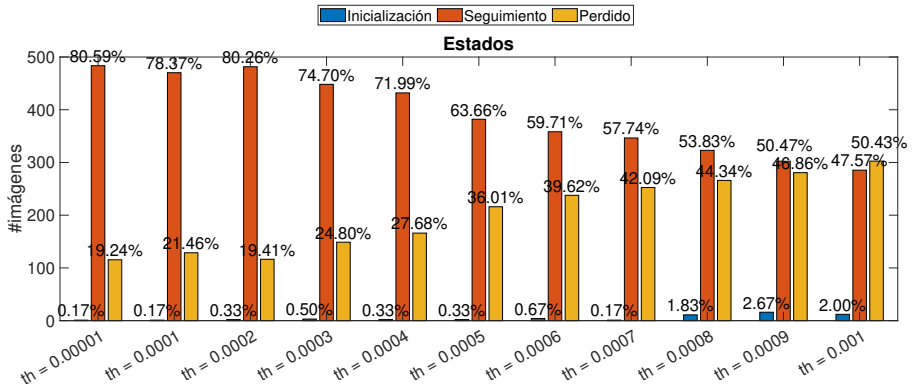
[15] `https://github.com/pablofdezalc/akaze`, último *commit* `25d5897` el 28 Oct 2016



(a) Tiempo medio de procesado.



(b) Puntos característicos.



(c) Número medio de imágenes en cada estado.

Figura 4.13: Evaluación del umbral (*threshold, th*) de A-KAZE.

puntos característicos, se mantiene un alto porcentaje de puntos del mapa emparejados. Esto se debe a que conforme se aumenta este umbral, se detectarán menos puntos pero éstos serán más robustos. Aunque también se dejarán de detectar muchos puntos de interés. En este caso es importante encontrar el balance para no detectar puntos en el ruido de la imagen o no detectar falsos positivos y detectar un número alto de puntos que contengan suficiente información sin eliminar puntos que son de interés (falsos negativos).

Por último, en la figura 4.13(c) se puede ver que a partir de $2 \cdot 10^{-4}$ el porcentaje de mapa estimado es muy similar. Si se tiene en cuenta que conforme disminuye el umbral aumenta el tiempo, el umbral $2 \cdot 10^{-4}$ es el que más se adecúa a este sistema ya que consigue estimar un mayor porcentaje de la trayectoria, en un tiempo muy razonable (90 ms) y consiguiendo emparejar un gran porcentaje de los puntos detectados con los puntos del mapa (18 %).

Una vez encontrado el umbral con el que se obtienen mejores resultados ($2 \cdot 10^{-4}$) se han evaluado los niveles y subniveles del espacio de escala no lineal. En la figura 4.14 se puede ver que el tiempo aumenta conforme aumenta el número niveles (n) y número subniveles (sn). Esto es debido a que se tienen que calcular un espacio de escala no lineal de mayor tamaño y que se debe buscar el máximo entre más escalas. Además, en la figura 4.15 se puede observar que con 3 niveles y 4 subniveles se pueden obtener resultados similares que con 3 niveles y 8 subniveles pero en menor tiempo.

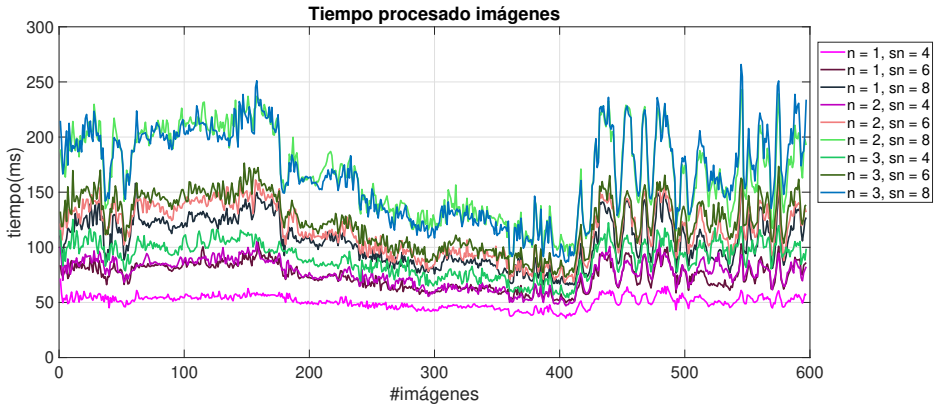


Figura 4.14: Tiempo medio de procesado de imagen según el número de niveles y subniveles.

En este apartado se puede concluir que modificando los parámetros del detector A-KAZE se puede disminuir el tiempo y aumentar el porcentaje de trayectoria estimada de la combinación A-KAZE-ORB. Además, al añadir el filtrado de puntos de los dos primeros subniveles se ha conseguido aumentar el porcentaje y disminuir el tiempo en 60 ms. Los mejores resultados se obtienen trabajando con A-KAZE-ORB con un umbral de $2 \cdot 10^{-4}$, con 3 niveles, 4 subniveles y aplicando el filtrado de puntos. De esta forma se consigue estimar un 80 % de la trayectoria con un tiempo de cómputo de 90 ms (11

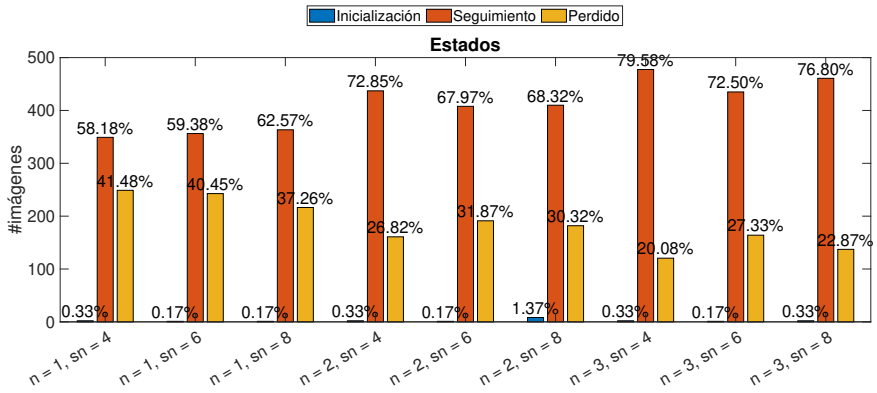


Figura 4.15: Número medio de imágenes en cada estado según el número de niveles y subniveles.

fps).

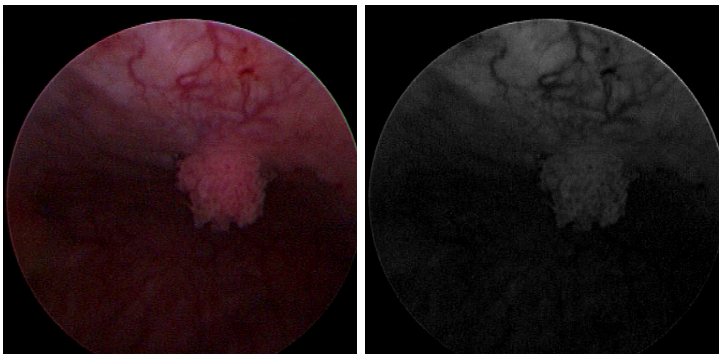
4.2.5. Pre-procesado de imagen

Otra posible forma de estimar un mayor porcentaje de la trayectoria es modificar los parámetros de CLAHE. CLAHE es el algoritmo utilizado para aumentar el contraste de las imágenes (sección 3.3). Este algoritmo solo necesita dos parámetros:

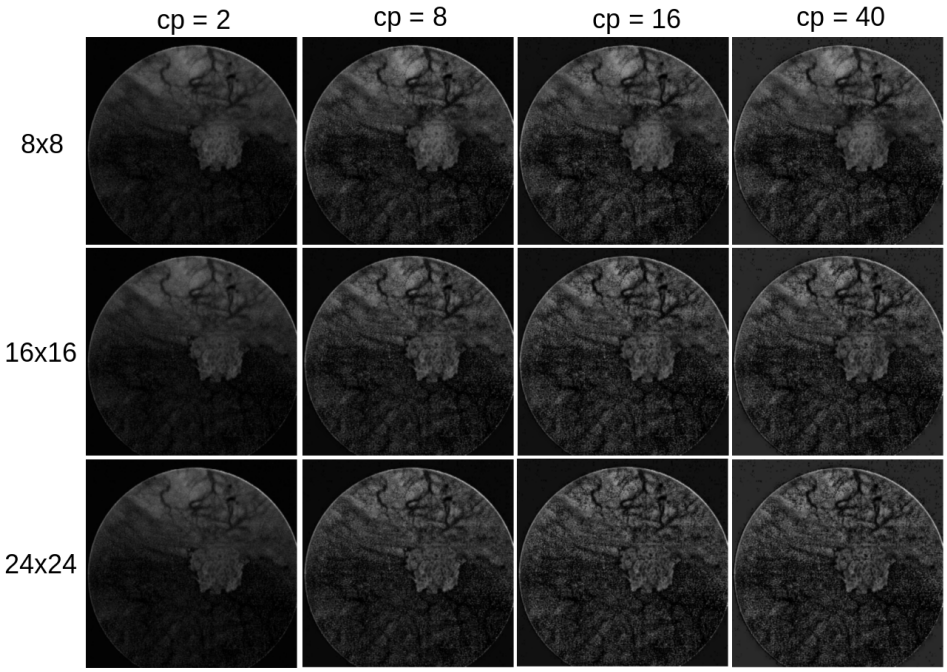
- Umbral o *clip limit*: usado para "cortar" las cajas del histograma antes de realizar la ecualización del histograma.
- Número de "tiles" o tamaño de malla. Como ya se ha mencionado CLAHE realiza un aumento del contraste de forma adaptativa. Para ello se ecualiza el histograma de pequeños bloques no sobrepuestos de la imagen. El número de bloques por fila y columna depende de este parámetro.

El impacto del cambio de estos parámetros sobre la imagen se puede ver en la figura 4.16. En esta figura se puede ver que conforme aumenta el umbral aumenta el ruido, ya que al aumentar el umbral CLAHE sería más similar a AHE y en AHE si hay ruido, éste se amplifica. El impacto del aumento del número de "tiles" apenas se puede apreciar ya que la calidad de la imagen depende principalmente del parámetro *clip limit* (CL) [39].

Para evaluar qué valores de los parámetros de CLAHE consiguen que mejore el funcionamiento de A-KAZE-ORB, se ha evaluado tanto el tiempo como la robustez de los puntos encontrados y el número de imágenes en las que el sistema está en el estado de seguimiento. En la figura 4.17(a) se puede observar que el cambio de parámetros apenas afecta al tiempo. En la figura 4.17(b) se puede observar que conforme aumenta el umbral aumenta el número de puntos detectados. Sin embargo, el número de puntos emparejados es bastante similar en todos los casos. Esto indica que los puntos que



(a) Imagen original. (b) Canal verde.

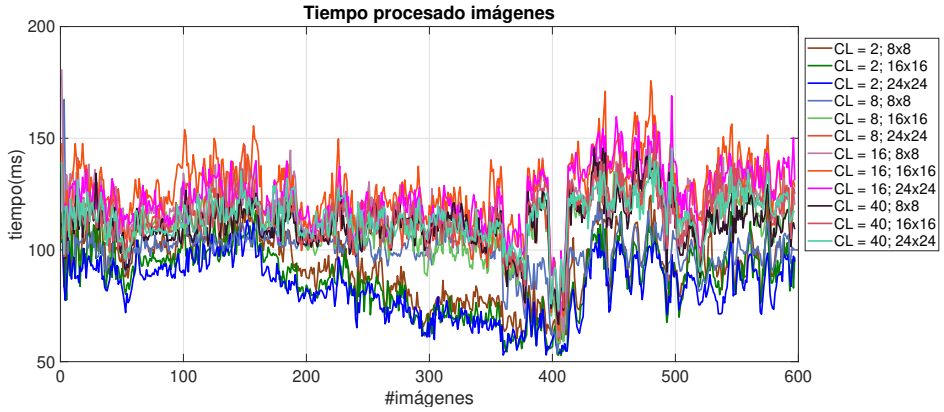


(c) Cambios en la imagen dependiendo de los parámetros de CLAHE (CL = *clip limit*).

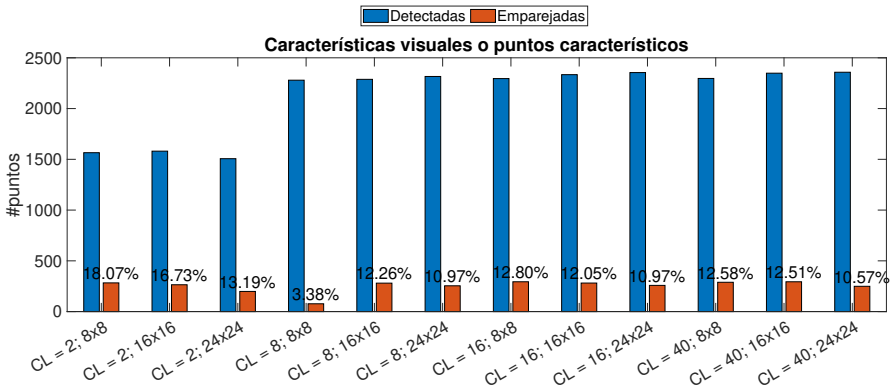
Figura 4.16: Impacto de los parámetros de CLAHE.

se detectan de más cuando el umbral es más alto, se detectan, probablemente, en el ruido de la imagen. Por último, en la figura 4.17(c) se puede observar que con un tamaño de malla (o número de bloques) de 8×8 y con un umbral o *clip limit* con valor 2, se consigue estimar la trayectoria y la posición de la cámara durante casi un 80 % de ésta.

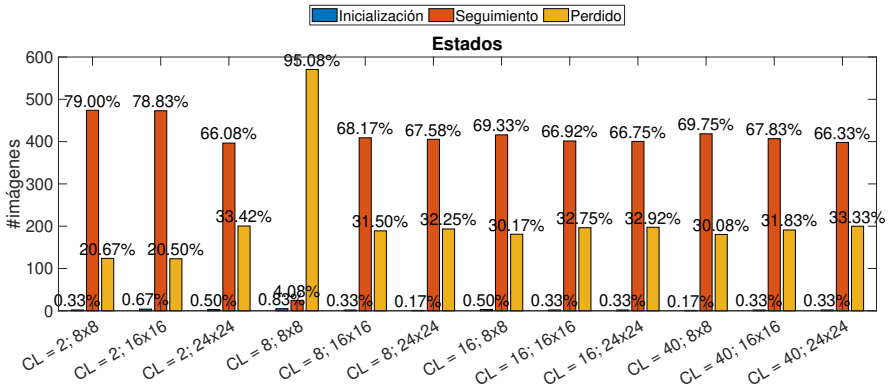
Una vez obtenidos los mejores valores para los parámetros de CLAHE y A-KAZE-BRIEF, se puede realizar una evaluación del pre-procesado de la imagen para ver cómo afecta éste al funcionamiento del sistema. Para ello se van a evaluar los siguientes pre-procesados de imagen:



(a) Tiempo medio de procesado de imagen.



(b) Puntos característicos.

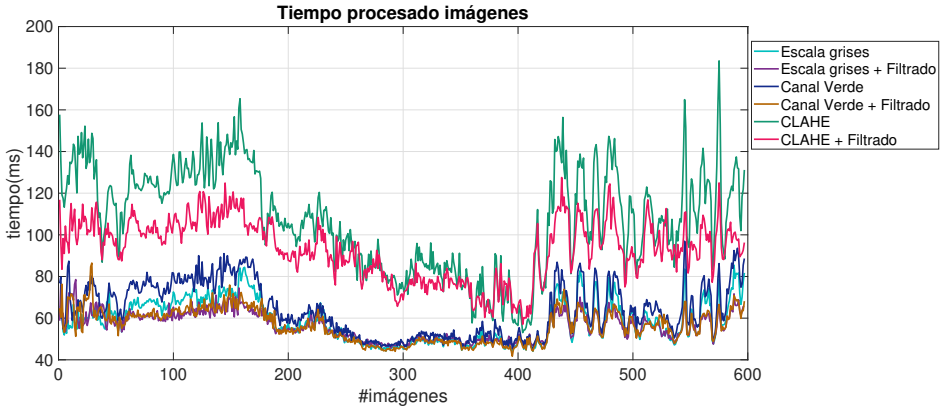


(c) Número medio de imágenes en cada estado.

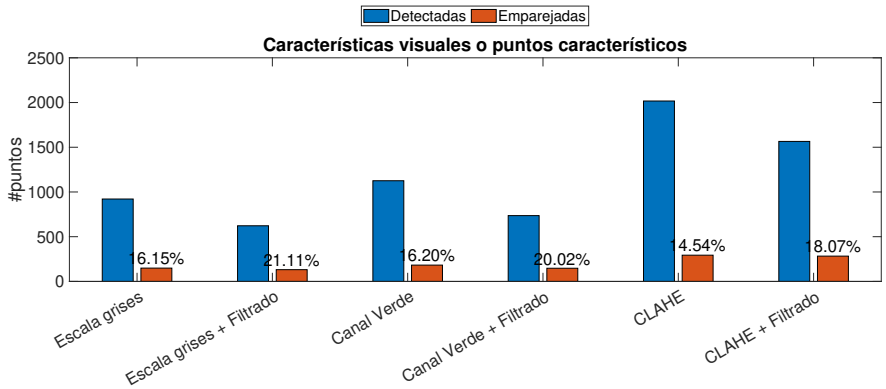
Figura 4.17: Evaluación de los parámetros de CLAHE: umbral (*clip limit, CL*) y tamaño de bloque ($pixels \times pixels$).

1. Imagen en escala de grises
2. Imagen en escala de grises y filtrado de puntos (ignorar los puntos de las dos primeras subescales)
3. Canal verde
4. Canal verde y filtrado de puntos
5. Canal verde de la imagen tras aplicar CLAHE
6. Canal verde de la imagen tras aplicar CLAHE y filtrado de puntos

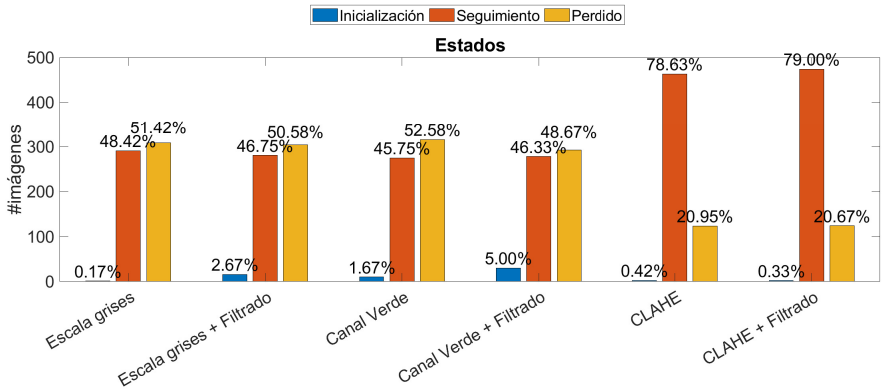
La evaluación de estos pre-procesados de imagen se puede observar en la figura 4.18. En la figura 4.18(a) se puede observar el tiempo de procesado de cada imagen (tiempo pre-procesado y detección de puntos de interés y seguimiento). En esta figura se puede observar que lo más caro es aplicar CLAHE y lo más barato es escoger el canal verde de la imagen. Además, se puede observar que al realizar el filtrado de puntos el tiempo disminuye. Esto es debido a que se han de detectar puntos en menos escalas. Además, en la figura 4.18(b) se puede observar que conforme más se procesa la imagen más puntos se detectan, pero estos puntos son más robustos ya que se consigue realizar un mayor número de emparejamientos con los puntos del mapa. Esto también se puede ver reflejado en la figura 4.18(c) ya que, conforme más emparejamientos se consiguen, mayor parte de la trayectoria se estima con éxito consiguiendo estimar cerca de un 80 % de la trayectoria.



(a) Tiempo medio de procesado de cada imagen.



(b) Puntos característicos.



(c) Número medio de imágenes en cada estado.

Figura 4.18: Evaluación del pre-procesado de imagen.

4.2.6. Interfaz

Un sistema VSLAM en una interfaz puede resultar muy útil ya que permite relacionar la imagen con el mapa 3D. Un punto seleccionado en la imagen puede ser convertido a un punto 3D del mapa estimado. Esto permite insertar anotaciones de realidad aumentada en cualquier punto del mapa. En una interfaz quirúrgica esto puede ser muy útil ya que permite al cirujano añadir anotaciones donde vea que le puede resultar útil. Además, estas anotaciones se guardan junto con el mapa 3D estimado de forma que, cuando el cirujano vuelva a pasar con el endoscopio por esa zona, la anotación seguirá allí.

Como ya se ha mencionado, en este tipo de operaciones el cirujano encuentra dificultades a la hora de diferenciar entre las zonas visitadas y las zonas que aún tiene que explorar. Para solucionar este problema, se podrían proyectar los puntos del mapa sobre la imagen creando una malla que indicara al cirujano si es una zona visitada. Otra opción sería que el cirujano tuviera la opción de insertar objetos virtuales 3D en ciertas partes del órgano.

Además, otro problema que encuentran los cirujanos es la pérdida de la orientación. Para ello podría insertarse en la imagen un objeto virtual indicando la orientación del endoscopio en forma de brújula (en la que el norte podría estar determinado por la cabeza del paciente) o mostrar la inclinación respecto una inclinación relativa (como por ejemplo, la inclinación del primer *keyframe*).

Además, en el caso de operaciones como la extracción de piedras en el riñón o la eliminación de pólipos en la vejiga, se podría añadir un módulo de segmentación para detectar estas estructuras en el interior del órgano y marcarlas en el mapa. De esta forma se podría indicar al cirujano dónde están las estructuras que hay que eliminar o qué estructura está extrayendo (ha de tenerse en cuenta que para eliminar un pólipo o una piedra es necesario realizar varias extracciones del endoscopio para eliminarla por completo).

Una interfaz que use un sistema VSLAM también puede ser muy útil en operaciones en las que se usan brazos robóticos manejados por el cirujano para operar. En este caso el cirujano puede seleccionar en la pantalla de forma precisa dónde quiere que el brazo robótico opere.

Además, para ejecutar una interfaz de este estilo sólo es necesario un ordenador que estuviera conectado a la salida de vídeo del endoscopio. Por lo que el precio del sistema apenas afectaría comparando el precio de un ordenados a los elevados precios de un endoscopio.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

En este trabajo se han realizado diferentes modificaciones a ORB-SLAM2 para conseguir estimar la trayectoria del endoscopio y el mapa 3D del órgano que se está examinando en operaciones de cirugía mínimamente invasiva. En este tipo de operaciones se encuentran varios problemas como la falta de textura, estructuras deformables, dificultad para encontrar puntos de interés fiables, movimientos bruscos y cambios de iluminación bruscos.

Se han realizado las siguientes modificaciones:

- Se ha modificado el sistema para que trabaje con menos puntos debido al pequeño campo de visión de la cámara.
- Se ha modificado la inicialización de ORB-SLAM2 para obtener un mapa inicial más robusto.
- Se ha añadido un pre-procesado de imagen al sistema para aumentar el contraste de las imágenes, disminuir el ruido de éstas y evitar detectar puntos en los reflejos causados por la iluminación artificial del endoscopio.
- Se ha modificado el detector y descriptor usado por ORB-SLAM2 para obtener puntos más robustos y emparejamientos más fuertes. Concluyendo que con A-KAZE-ORB se consigue estimar un mayor porcentaje del mapa.
- Se han creado nuevos vocabularios usando las imágenes de la base de datos *Kvasir*.

Cambiando la combinación de detector y descriptor de FAST-ORB a A-KAZE-ORB se ha conseguido aumentar el porcentaje de trayectoria estimada a un 80 % en una secuencia de eliminación de pólipos de la vejiga. Con el sistema original no se llegaba a estimar el 50 % de la misma secuencia.

Además, este cambio apenas a afectado al tiempo, se pasa de 50 ms con FAST-ORB a 90 ms con A-KAZE-ORB. Con A-KAZE-A-KAZE también se han obtenido muy buenos resultados pero a cambio el tiempo ha aumentado a aproximadamente 250 ms, por lo que esta combinación no funcionaría en tiempo real. Lo que indica que el descriptor ORB es bastante más barato de calcular que el descriptor A-KAZE pero con ambos descriptores se obtienen resultados similares. Dado que el descriptor ORB es más barato, la combinación A-KAZE-ORB es más apto para aplicaciones en tiempo real, manteniendo la detección de puntos robustos de A-KAZE-A-KAZE.

En conclusión, con la mejoras realizadas en ORB-SLAM2 con respecto a los diferentes umbrales, incialización y pre-procesado de imagen se ha conseguido aumentar el porcentaje de trayectoria estimada con respecto al sistema original. Sin embargo, con estas modificaciones no se llegaba a estimar más del 40 % de la trayectoria debido al detector de puntos FAST (combinación FAST-ORB). Por este motivo, se decidió probar diferentes combinaciones obteniendo que usando A-KAZE-ORB con un filtrado de puntos en los dos primeros subniveles de la imagen, el pre-procesado de la imagen (CLAHE), un umbral de $2 \cdot 10^{-4}$ y con un espacio de escala no lineal de 3 niveles y 4 subniveles se llega a estimar un 80 % de la trayectoria tardando en procesar unos 90 ms cada imagen.

Se ha de resaltar que estos tiempos han sido medidos en un ordenador portátil y es de esperar que este tiempo se reduzca usando un ordenador de sobremesa o clusters de computación en un sistema de cirugía mínimamente invasiva.

5.2. Trabajo futuro

Este trabajo se ha demostrado que modificando ORB-SLAM2 se puede obtener un sistema de SLAM capaz de estimar el mapa 3D del entorno y la trayectoria del endoscopio en operaciones de cirugía mínimamente invasiva. Este sistema trabaja a 11 fps y consigue estimar un alto porcentaje de la trayectoria aunque sin llegar a estimar aún el 100 % de ésta. Debido a esto existen diferentes líneas futuras de investigación en este proyecto que planeamos llevar a cabo en el mismo laboratorio:

- Disminución del tiempo de cómputo del sistema de A-KAZE-A-KAZE y A-KAZE-ORB implementándolo en GPU.
- Aumento de la robustez del sistema frente a los movimientos bruscos de la cámara.
- Evaluación del cierre de bucle.
- Obtención de *datasets* más amplios para la creación de un nuevo vocabulario.
- Extensión del sistema a otros procedimientos quirúrgicos como por ejemplo en laparoscopia abdominal.

- Implementación de una interfaz que utilice este sistema y que ayude a los cirujanos en las operaciones de cirugía mínimamente invasiva.
- Obtención de más *datasets* correspondientes a operaciones de cirugía mínimamente invasiva.
- Evaluación del sistema y de la interfaz en los riñones sintéticos texturizados.
- Detección y reconocimiento de las estructuras a extraer o eliminar (como pólipos o piedras).
- Estudio de características aprendidas por *Machine Learning*.

Bibliografía

[1] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007.

[2] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[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] 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*, 33(5):1255–1262, 2017.

[5] Mark Cummins and Paul Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[6] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[7] Jianbo Shi. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE, 1994.

[8] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[10] Pablo Fernández Alcantarilla. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011.

[11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

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

[13] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[14] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[15] Iñigo Azqueta-Gavaldon, Florian Fröhlich, Klaus Strobl, and Rudolph Triebel. Segmentation of surgical instruments for minimally-invasive robot-assisted procedures using generative deep neural networks. *arXiv preprint arXiv:2006.03486*, 2020.

[16] Chauncey Graetzel, Terry Fong, Sebastien Grange, and Charles Baur. A non-contact mouse for surgeon-computer interaction. *Technology and Health Care*, 12(3):245–257, 2004.

[17] Atsushi Nishikawa, Toshinori Hosoi, Kengo Koara, Daiji Negoro, Ayae Hikita, Shuichi Asano, Haruhiko Kakutani, Fumio Miyazaki, Mitsugu Sekimoto, Masayoshi Yasui, et al. FAce MOUSe: A novel human-machine interface for controlling the position of a laparoscope. *IEEE Transactions on Robotics and Automation*, 19(5):825–841, 2003.

[18] Oscar G Grasa, Ernesto Bernal, Santiago Casado, Ismael Gil, and JMM Montiel. Visual SLAM for handheld monocular endoscope. *IEEE transactions on medical imaging*, 33(1):135–146, 2013.

[19] Nader Mahmoud, Iñigo Cirauqui, Alexandre Hostettler, Christophe Doignon, Luc Soler, Jacques Marescaux, and JMM Montiel. ORBSLAM-based endoscope tracking and 3D reconstruction. In *International workshop on computer-assisted and robotic endoscopy*, pages 72–83. Springer, 2016.

[20] Nader Mahmoud, Alexandre Hostettler, Toby Collins, Luc Soler, Christophe Doignon, and JMM Montiel. SLAM based quasi dense reconstruction for minimally invasive surgery scenes. *arXiv preprint arXiv:1705.09107*, 2017.

[21] Ulrich Hagn, Rainer Konietschke, Andreas Tobergte, Mathias Nickl, Stefan Jörg, Bernhard Kübler, Georg Passig, Martin Gröger, Florian Fröhlich, Ulrich Seibold, et al. DLR MiroSurge: a versatile system for research in endoscopic telesurgery. *International journal of computer assisted radiology and surgery*, 5(2):183–193, 2010.

[22] K. H. Strobl, W. Sepp, S. Fuchs, C. Paredes, M. Smisek, and K. Arbter. DLR CalDe and DLR CalLab.

[23] Klaus H Strobl. *A flexible approach to close-range 3-D modeling.* PhD thesis, Technische Universität München, 2014.

[24] Shaharyar Ahmed Khan Tareen and Zahra Saleem. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*, pages 1–10. IEEE, 2018.

[25] Joachim Weickert, BM Ter Haar Romeny, and Max A Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE transactions on image processing*, 7(3):398–410, 1998.

[26] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[27] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. KAZE features. In *European Conference on Computer Vision*, pages 214–227. Springer, 2012.

[28] Xin Yang and Kwang-Ting Cheng. Ldb: An ultra-fast feature for scalable augmented reality on mobile devices. In *2012 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 49–57. IEEE, 2012.

[29] Lester Kalms, Khaled Mohamed, and Diana Göhringer. Accelerated embedded akaze feature detection algorithm on fpga. In *Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies*, pages 1–6, 2017.

[30] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[31] Karel Zuiderveld. Contrast limited adaptive histogram equalization. *Graphics gems*, pages 474–485, 1994.

[32] Gabriel Fillipe Centini Campos, Saulo Martiello Mastelini, Gabriel Jonas Aguiar, Rafael Gomes Mantovani, Leonimer Flávio de Melo, and Sylvio Barbon. Machine learning hyperparameter selection for contrast limited adaptive histogram equalization. *EURASIP Journal on Image and Video Processing*, 2019(1):59, 2019.

[33] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. KVASIR: A multi-class image dataset for

computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, pages 164–169, New York, NY, USA, 2017. ACM.

[34] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[35] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. page 1470. IEEE, 2003.

[36] Dorian Gálvez-López and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.

[37] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.

[38] KP Horn Berthold and P Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the optical society of America*, 4(4):629–642, 1987.

[39] Byong Seok Min, Dong Kyun Lim, Seung Jong Kim, and Joo Heung Lee. A novel method of determining parameters of clahe based on image entropy. *International Journal of Software Engineering and Its Applications*, 7(5):113–120, 2013.

Índice de figuras