Universidad Zaragoza
1542

Trabajo Fin de Grado

Reconocimiento de lugares en SLAM visual con imágenes de endoscopio

Place recognition in Visual SLAM with endoscopic sequences

Autor

Óscar Pueyo Ciutad

Directores

Juan Domingo Tardós Solano

Juan José Gómez Rodríguez

Titulación

Grado en Ingeniería en Informática

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2022

AGRADECIMIENTOS

Agradecer en primer lugar tanto a Juan Domingo Tardós como a Juan José Gómez por su ayuda y orientación a lo largo de todo el proyecto, siempre dispuestos a guiarme en mi inicio en el mundo de la investigación.

También me gustaría agradecer a mi novia, familiares, amigos y compañeros de promoción que me han acompañado y apoyado durante todos estos años de grado.

Finalmente, agradecer al Ministerio de Educación y Formación Profesional por la Beca de Colaboración que me ha permitido formar parte de este proyecto.

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

RESUMEN

El SLAM (*Simultaneous Localization and Mapping*) Visual es un problema computacional que consiste en la construcción y actualización de un mapa en un entorno desconocido mientras simultáneamente se actualiza la posición de la cámara dentro de él, tomando únicamente como entrada las imágenes captadas por la cámara. Este trabajo de fin de grado trata el problema de SLAM topológico basado en apariencia, donde el mapa está formado por un grafo de lugares con su descripción visual, en el interior del colon. Las técnicas clásicas asumen que el entorno es rígido, mientras que en imágenes de endoscopia el escenario es deformable. Además, se añaden problemas como la escasa textura del colon frente a la presente en el mundo exterior, las continuas oclusiones, la iluminación cambiante y los reflejos especulares.

Primero, se ha desarrollado un método de reconocimiento de lugares basado en bolsas de palabras visuales a partir de puntos de interés AKAZE. Se ha realizado una máscara de brillos para ignorar los reflejos especulares, un filtro de imágenes borrosas, y se ha ajustado AKAZE para colonoscopias, utilizando su implementación en GPU para que sea viable en tiempo real. El método ha sido sintonizado y evaluado en secuencias de colonoscopia del proyecto europeo EndoMapper. Los resultados obtenidos logran un *recall* (fracción de imágenes que se reconocen correctamente) del $57 \sim 78\,\%$ sin ningún falso positivo, siendo un método muy adecuado para relocalización en SLAM.

Segundo, la técnica de reconocimiento de lugares se ha integrado en un algoritmo de localización Bayesiana, que permite localizar la cámara dentro de un mapa topológico del interior del colon. Con un mapa previo construido a mano, la localización Bayesiana consigue mejorar el *recall* hasta un $78 \sim 92\,\%$.

Finalmente, se ha desarrollado un método de SLAM topológico basado en apariencia que, integrando el algoritmo de localización Bayesiana, es capaz de construir automáticamente el grafo de lugares que constituye el mapa topológico. El método se ha validado localizando el endoscopio con el mapa construido, alcanzando un *recall* del $90 \sim 99\,\%$.

En este enlace se pueden ver demostraciones de los distintos sistemas desarrollados.

Índice

Capítulo 1

Introducción y objetivos

1.1. Motivación

El problema de SLAM (*Simultaneous Location and Mapping*) consiste en la construcción y actualización de un mapa en un entorno desconocido mientras simultáneamente se localiza al robot en él. Existen diversas aproximaciones para este problema, pero en este proyecto se ha tratado el SLAM Visual (VSLAM), donde la única información usada para esta tarea son las imágenes de entrada.

El trabajo realizado se enmarca en el proyecto europeo EndoMapper, que trata de reconstruir mapas del interior del cuerpo humano mediante cámaras monoculares en cirugía mínimamente invasiva. El colon sufre continuas deformaciones tanto en aspecto como en estructura. Sin embargo, las técnicas actuales sólo funcionan en entornos rígidos. Investigar técnicas de reconocimiento de lugares en secuencias de endoscopio, que proporcionan un entorno de investigación no rígido, permitirá recuperar la localización de la cámara tras oclusiones, y emparejar imágenes obtenidas a la entrada y a la salida del aparato para guiar al cirujano. El desarrollo de un sistema de VSLAM facilitaría la navegación del cirujano en el colon, permitiendo añadir etiquetas a las distintas zonas.

El colon plantea problemas adicionales, como la falta de textura en las paredes, las continuas oclusiones que sufre la cámara, los cambios de iluminación y los brillos especulares, por lo que un sistema robusto frente a esta casuística aumentaría la calidad de los métodos usados en estas intervenciones.

Este trabajo ha sido realizado con la ayuda de la Beca de Colaboración del Ministerio de Educación.

1.2. Trabajo previo

En trabajos previos [1] se modificó el ORBSLAM2 [2] sustituyendo la extracción-descripción de puntos de interés FAST-ORB a una combinación AKAZE-ORB y junto a un nuevo vocabulario se logró mejorar el porcentaje de imágenes en el que el sistema se mantiene en seguimiento de un 40 % a un 80 % en cirugía uretral mínimamente invasiva.

Las características artesanales (SIFT [3], ORB [4] y AKAZE [5]) se compararon con el estado del arte de redes neuronales (SuperPoint [6] y SuperGlue [7]) para reconocimiento de lugares en el colon [8]. Se observó que las mejores características artesanales eran AKAZE, que alcanzaban un *recall* de 61,47 %, mientras que las redes neuronales obtenían un *recall* de 70,69 % que es ligeramente superior, a costa de introducir una mayor complejidad en el sistema. Por este motivo en este trabajo se va a profundizar en la investigación de técnicas de reconocimiento de lugares en colonoscopias usando AKAZE.

1.3. Objetivos

El objetivo de este proyecto es desarrollar un sistema de relocalización y SLAM que logre buenos resultados en entornos deformables, en este caso en el interior del colon. Para ello, se van a seguir los siguientes pasos:

- Estudio y diseño de un método de reconocimiento de lugares con bolsas de palabras binarias [9] con puntos AKAZE [10, 5].
- Sintonización de AKAZE para mejorar su comportamiento en imágenes del colon.
- Pruebas de reconocimiento de lugares en secuencias de EndoMapper.
- Estudio y diseño de un sistema de SLAM topológico basado en apariencia.
- Evaluación del sistema con distintos parámetros.
- Elaboración de un prototipo demostrador del sistema con secuencias de EndoMapper.

1.4. Metodología y herramientas

El proyecto consiste en el desarrollo y evaluación de los métodos mencionados, reusando lo máximo posible las partes ya desarrolladas. El desarrollo del sistema se realiza de forma iterativa, donde tras el desarrollo de cada módulo se prueba y ajusta

para verificar que su comportamiento es el esperado. Las datos, librerías y herramientas utilizadas son:

- EndoMapper Dataset [11]: Conjunto de grabaciones de endoscopias sobre las que se prueba el sistema.
- OpenCV: Librería de código abierto de visión por computador.
- AKAZE [10, 5]: Librería de detección y extracción de puntos AKAZE.
- CUDA_AKAZE [12]: Librería de detección y extracción de puntos AKAZE en GPU mediante CUDA.
- DBoW3 [13]: Librería de índice invertido, vocabulario de palabras visuales y creación y comparación de bolsas de palabras.
- C++: Lenguaje de programación del sistema.
- CMake: Herramienta de generación y automatización de código.
- CLion: Entorno de desarrollo para C++ y CMake.
- Git y GitHub: Control de versiones del código y guardado en la nube.
- MATLAB: Lenguaje de programación de matrices, para reconstrucción de escenas en pruebas, generación de gráficas y visualización de resultados.
- Python: Evaluación y visualización de resultados.
- Bash: Lenguaje de *scripting*, para automatización de pruebas y obtención de resultados.
- Overleaf (LaTeX): Software para redacción de documentos.
- diagrams.net: Creación de diagramas.
- GIMP y Paint.NET: Edición de imagen.

1.5. Descripción del documento

Tras la actual introducción, en el capítulo 2 se explican los fundamentos previos de visión por computador necesarios para comprender el proyecto realizado. En el capítulo 3 se detalla la sintonización del detector AKAZE para reconocimiento de lugares en endoscopias, el procesado de imagen, la máscara generada y el algoritmo de

emparejamiento. En el capítulo 4 se trata el problema de reconocimiento de lugares en endoscopias, la selección de candidatos y la verificación de estos. En el capítulo 5 se diseña el método de localización del endoscopio dentro de un mapa topológico mediante técnicas Bayesianas y el método de SLAM topológico. Finalmente, se muestran las conclusiones del proyecto y el posible trabajo futuro.

Capítulo 2

Fundamentos previos

2.1. Puntos de interés

En visión por computador, los puntos de interés son zonas repetibles y distinguibles que representan una cierta región de la imagen. Estos puntos son asociados normalmente a esquinas, puesto que en los contornos el punto se podría desplazar a lo largo de este, siendo difícilmente distinguibles. A lo largo de la historia se han desarrollado distintas técnicas para abordar este problema que se dividen principalmente en dos tipos: características artesanales y *deep learning* (aprendizaje profundo).

Las características artesanales son aquellas cuyo algoritmo de detección y extracción están desarrollados de forma completamente manual y sus propiedades derivan de dicho algoritmo, mientras que las basadas en aprendizaje profundo usan redes neuronales para aprender tanto la detección como descripción de estos puntos.

2.2. AKAZE

AKAZE [10, 5] es una técnica publicada en 2013 para detección y extracción de puntos de interés en espacios de escala no lineales. Construye una pirámide de escalas (ver Figura 2.1) de imagen en la que en cada nivel reduce el tamaño y realiza progresivamente un proceso de difusión que respeta los contornos de los objetos y suaviza la imagen, tratando de eliminar el ruido y enfatizar detalles y contornos. Sobre cada una de estas imágenes se realiza posteriormente la detección y descripción de los puntos de interés.

2.2.1. Filtro de difusión no lineal

La difusión es un proceso que trata de equilibrar la concentración de gradientes de la imagen mediante un flujo. La ecuación 2.1 muestra la formulación del problema [10,

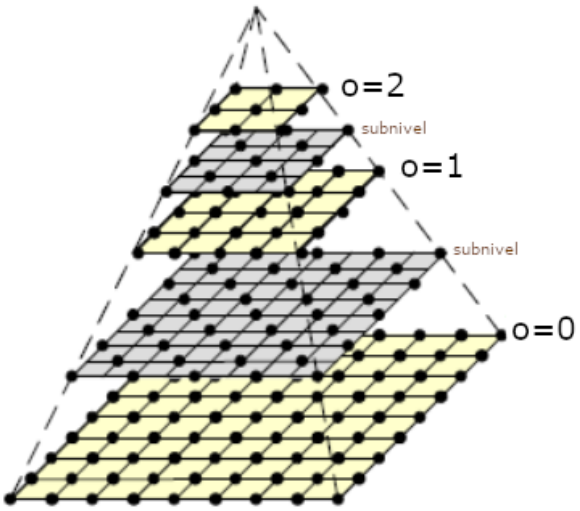


Figura 2.1: Pirámide de escala de una misma imagen. Cuanto mayor es el valor de o (octava), menor es el tamaño de la imagen debido a la reducción del tamaño a la mitad en cada una. Entre cada una de estas octavas hay S subniveles intermedios (S=1 en este ejemplo). La octava $o = 0$ es la imagen original.

14]:

$$\frac{\partial L}{\partial t} = div(c(x, y, t) \cdot \nabla L) = c(x, y, t) \cdot \nabla^2 L \tag{2.1}$$

donde div y ∇^2 es el operador de divergencia o Laplaciano (mide la cantidad de flujo que sale menos la que entra) y ∇ el gradiente (magnitud de cambio en la dirección del cambio). La idea es que cuando el gradiente es pequeño la difusión difumina y elimina ruido, mientras que si el gradiente es grande la difusión no actúa. Mediante la difusión se describe la evolución de la imagen a través de las escalas y subniveles de la pirámide.

La difusión adaptativa a la estructura local de la imagen se debe a la función de conductividad c, y t es el parámetro de escala (valores más grandes dan lugar a representaciones más simples):

$$c(x, y, t) = g(|\nabla L_\sigma(x, y, t)|) \tag{2.2}$$

donde ∇L_σ es el gradiente de la imagen (nótese que si el valor absoluto del gradiente de una zona de la imagen es alto significa que hay un detalle o contorno) suavizada con un kernel gaussiano.

Existen diversas funciones c que hacen uso de la magnitud del gradiente:

$$pm\_g1 = exp(-\frac{|\nabla L_\sigma|^2}{k^2}) \tag{2.3}$$

$$pm\_g2 = \frac{1}{1 + \frac{|\nabla L_\sigma|^2}{k^2}} \tag{2.4}$$

$$weick = \begin{cases} 1 & |\nabla L_\sigma|^2 = 0 \\ 1 - exp(-\frac{3{,}315}{(|\nabla L_\sigma|/k)^8}) & |\nabla L_\sigma|^2 > 0 \end{cases} \tag{2.5}$$

Resumidamente, la función pm_g1 [15] [1] (ec. 2.3) favorece bordes con alto contraste, mientras que la pm_g2 [15] (ec. 2.4) promueve regiones grandes frente a pequeñas. En la función de Weickert [16] (ec. 2.5) hay más suavizado en los dos lados del borde que en este, es decir, prefiere el suavizado intrarregional al difuminado interregional. La Figura 2.2 muestra el comportamiento de las distintas funciones de difusividad.

El parámetro k es un factor de contraste[2] que controla cuánto reacciona la función de conductividad, un valor mayor difuminará más la imagen, eliminando más bordes, mientras que si el valor es pequeño difuminará menos partes de la imagen (ver Figura 2.3).

El proceso de difusión se realiza en una pirámide de escala, para poder extraer puntos de interés en lugares con detalles más grandes, en un conjunto de O octavas y S subniveles:

$$\sigma_i(o, s) = 2^{o+\frac{s}{S}}, o \in [0...O-1], s \in [0...S-1], i \in [0...M] \tag{2.6}$$

donde M es el número total de imágenes filtradas. En la Figura 2.1 se puede observar una pirámide de escala con distintas octavas (reducciones del tamaño de la imagen entre 2^o). Los subniveles son puntos intermedios entre estas octavas. Este conjunto de imágenes se le denomina L, siendo cada una de estas imágenes L^i.

2.2.2. Detección de puntos de interés

Los puntos se obtienen calculando el determinante del Hessiano en cada una de las imágenes L^i. El conjunto de operadores están normalizados respecto a la escala, teniendo en cuenta la octava de cada imagen $\sigma_{i,norm} = \sigma_i/2^{\sigma^i}$ y

$$L^i_{Hessian} = \sigma^2_{i,norm}(L^i_{xx}L^i_{yy} - L^i_{xy}L^i_{xy}) \tag{2.7}$$

donde las derivadas se calculan con los filtros de Scharr en x e y (esta matriz de convolución se multiplica para cada ventana 3×3 para obtener el gradiente):

$$K_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \quad K_y = \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \tag{2.8}$$

Dados los puntos cuyo determinante del Hessiano ha superado un umbral preestablecido, se filtran los que no son un máximo local en una ventana de tamaño

[1] pm = Perona y Malik, autores de las funciones 1 y 2.

[2] Durante todo el documento se seguirá llamando k, pero no al valor final de contraste, sino al percentil k del histograma de gradientes de la imagen, que es el valor que finalmente se usa en la fórmula, para adaptarse a las propiedades de la imagen.

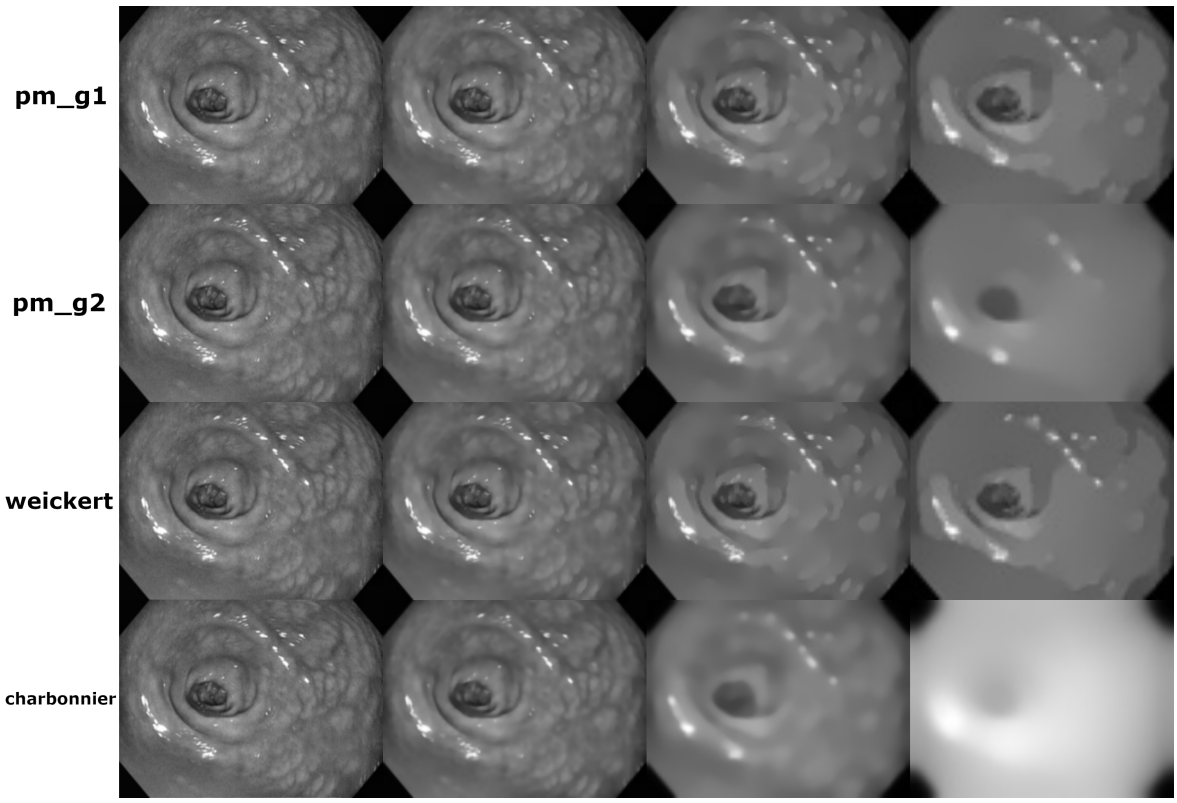


Figura 2.2: Comparación de las distintas funciones de difusividad a lo largo de cuatro octavas (entre octavas hay 4 subniveles).

3×3 (el valor es el máximo entre sus vecinos de la ventana) y se descartan los que la respuesta no es máxima en los niveles $i + 1$ e $i - 1$ (de la pirámide de escala) en una ventana de tamaño $\sigma_i \times \sigma_i$ píxeles.

2.2.3. Descripción de los puntos de interés

El descriptor de los puntos de interés es el *Modified-Local Difference Binary* (M-LDB), basado en el descriptor LDB, que usa tests binarios entre medias de áreas en vez de píxeles únicos por robustez. Además de tests sobre los valores de intensidad, también se añaden tests sobre la media de los gradientes horizontales y verticales en el área.

Para la invarianza a rotación, se estima la orientación dominante del área y se rota la rejilla de LDB (ver Figura 2.4).

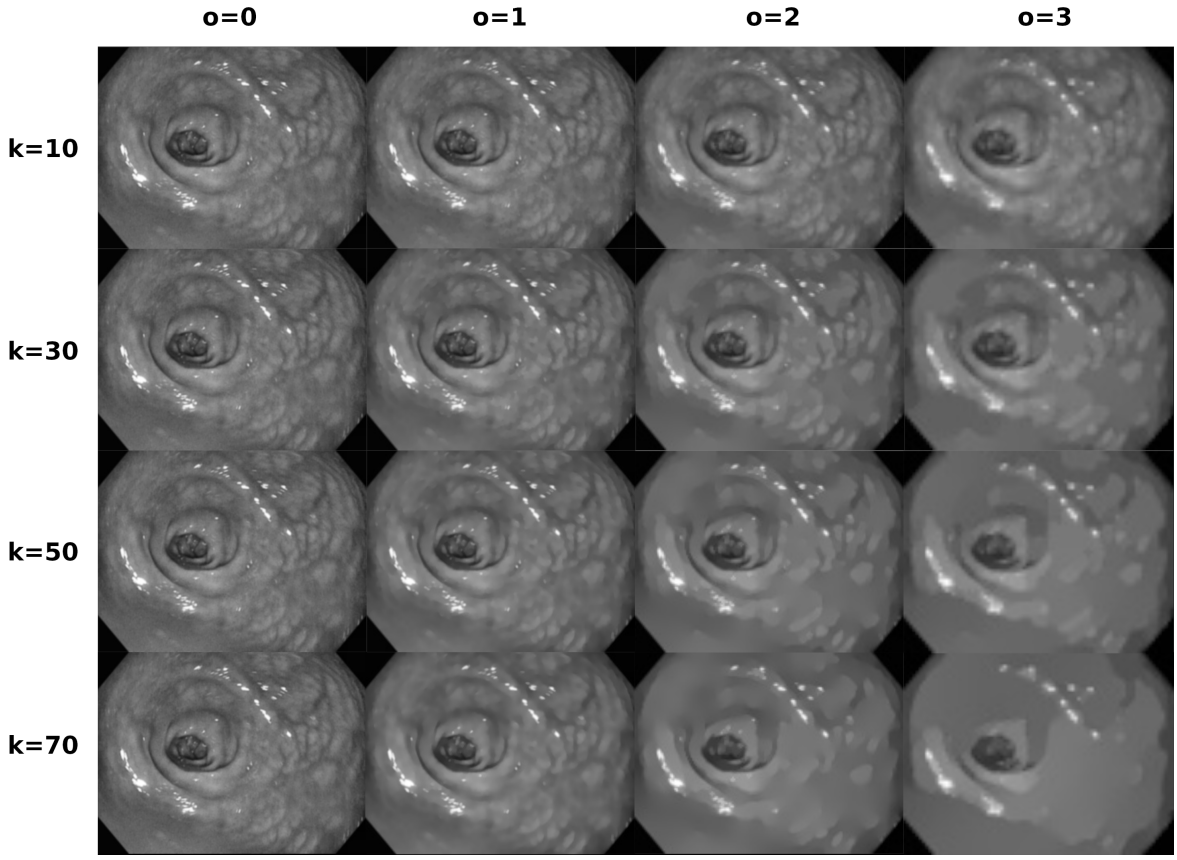
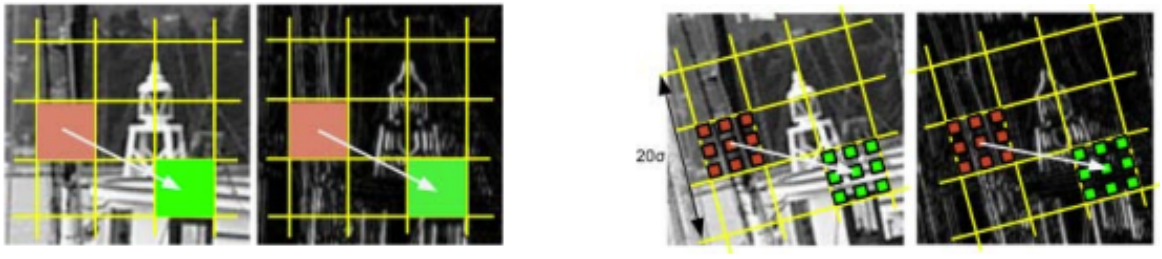


Figura 2.3: Evolución de la imagen según el parámetro de contraste k que controla la difusividad en las distintas octavas o y subescalas ($s = 0$ en todas las imágenes) usando la función de *weickert*.



(a) LDB (b) MLDB

Figura 2.4: Ejemplo de tests del descriptor LDB y MLDB [5]. En el descriptor MLDB la rejilla se rota para obtener invarianza a rotación. La comparación entre regiones se realiza comparando la media de la intensidad, la derivada en x y la derivada en y (3 bits por comparación).

2.3. Reconocimiento de lugares

El problema del reconocimiento de lugares es un tema muy estudiado en visión por computador y robótica, que consiste en reconocer precisa y eficientemente la localización de una imagen de consulta. Dada esta imagen de consulta y una base de datos de imágenes previas, el objetivo es encontrar la imagen anterior más parecida a esta.

2.3.1. Bolsas de palabras binarias para reconocimiento de lugares

Para detectar lugares revisitados la aproximación usada es una base de datos compuesta por un vocabulario visual jerárquico [9], tanto con un índice directo como con uno inverso (ver Figura 2.5).

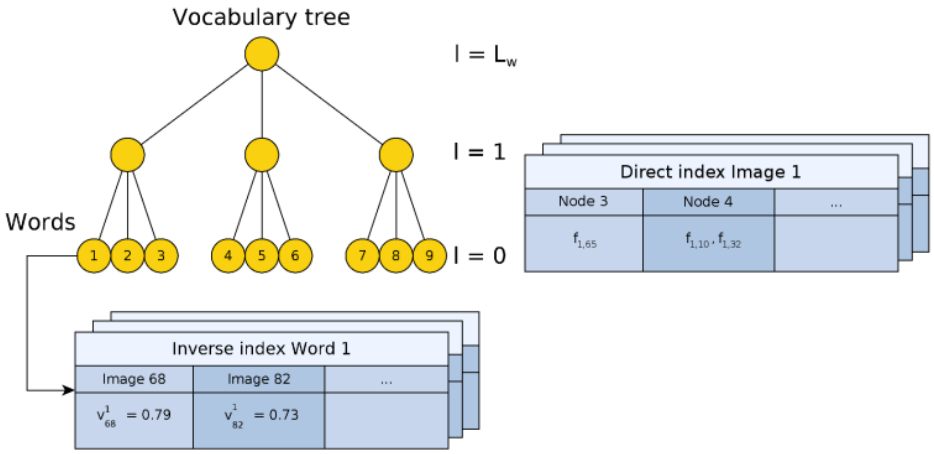


Figura 2.5: Árbol de vocabulario de ejemplo con índices directo e inverso de la base de datos de imágenes [9].

Esta técnica usa un vocabulario visual para convertir la imagen en un vector cuasi-vacío. Este vocabulario se crea discretizando el espacio del descriptor en W palabras visuales, estructurado en forma de árbol. Para construirlo se extraen puntos de interés durante una etapa de entrenamiento y se discretizan en k_w clusters binarios mediante *k-medians*. Este proceso se realiza recursivamente L_w veces hasta llegar a tener W hojas, que son las palabras visuales del vocabulario. Además, a cada hoja se le da un peso basado en su aparición en el corpus de entrenamiento usando el esquema de pesos $tf - idf$ (frecuencia de término - frecuencia inversa de documento).

Para transformar una imagen I a una bolsa de palabras, se extraen puntos de interés y, con sus descriptores, cada uno recorre el árbol desde la raíz hasta las hojas, seleccionando en cada nivel el nodo que minimiza la distancia de Hamming[3].

[3]En nuestro caso puesto que los descriptores de AKAZE son binarios.

Para medir la similitud entre dos vectores bolsas de palabras se usa la norma L_1:

$$s(v_1, v_2) = 1 - \frac{1}{2}\left|\frac{v_1}{|v_1|} - \frac{v_2}{|v_2|}\right| \tag{2.9}$$

El índice inverso guarda para cada palabra w_i una lista de imágenes I_t donde aparece con el objetivo de comparar únicamente la imagen consulta con las imágenes que comparten alguna palabra en común. Por último, el índice directo sirve para guardar los puntos de interés de cada imagen. Para esto se separan los nodos del árbol del vocabulario según su nivel l, empezando desde las hojas con $l = 0$ hasta la raíz $l = L_w$. Para cada imagen I_t se guarda en el índice directo en los nodos del nivel l que son ancestros de las palabras presentes en I_t junto a los puntos de interés asociados a cada nodo. Esto sirve para aproximar el vecino más cercano y acelerar la verificación geométrica.

La base de datos, por tanto, sirve para recuperar eficientemente imágenes similares a una imagen consulta. Esta imagen consulta, transformada a vector de bolsa de palabras se busca en la base de datos, dando lugar a una lista de candidatos con su puntuación asociada.

2.4. Verificación geométrica

Dados unos emparejamientos de puntos de interés entre 2 imágenes, la matriz esencial es un método para verificar la consistencia geométrica de estos, filtrando espurios para tener un conjunto final robusto. Esta técnica consiste en usar la geometría de 2 vistas para triangular puntos en una escena y ver si la reproyección a los planos de imagen es la correcta. La ecuación de la verificación de un punto es la siguiente:

$$x_{c1}^T E x_{c0} = 0 \tag{2.10}$$

donde x_{c1}^T representa el punto x observado desde el centro óptico c_1 traspuesto (ver Figura 2.6) y x_{c0} es la observación del punto x desde el centro óptico c_0.

Este método consiste en hallar el desplazamiento T y rotación R entre los centros ópticos $c_{\{0,1\}}$ (que dan lugar a la matriz esencial E), triangulando los puntos de las 2 vistas 2D al 3D. La triangulación de un punto ocurre donde cortan los dos vectores $\vec{v}_{c_0,x_0}$ y $\vec{v}_{c_1,x_1}$. La matriz esencial (que relaciona las vistas desde c_0 y c_1) se obtiene mediante los emparejamientos del paso anterior, siguiendo un esquema de RANSAC [17] para evitar calcular el modelo con espurios, que lo degeneraría. La matriz esencial E se calcula normalmente mediante el algoritmo de 8 puntos [18], puesto que aunque solo se necesiten 5 su implementación no es práctica y requiere resolver ecuaciones no

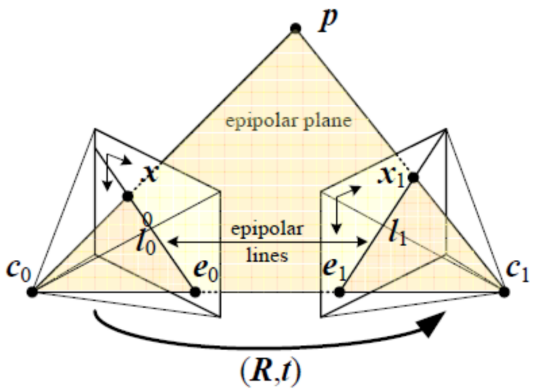


Figura 2.6: Restricción epipolar a partir de 2 vistas de un punto x dados los planos de imagen observados desde los centros ópticos c_1 y c_0.

lineales. Usando E, verificar que el punto p es consistente geométricamente consiste en medir el error angular mediante:

$$|cos^{-1}(\frac{E * \vec{v}_{c_0,x_0}}{\|E * \vec{v}_{c_0,x_0}\|} * \frac{\vec{v}_{c_1,x_1}}{\|\vec{v}_{c_1,x_1}\|})| < \epsilon \tag{2.11}$$

donde ϵ es el error angular (entre el punto y la línea epipolar l_1) y $E * \vec{v}_{c_0,x_0}$ es el vector $\vec{v}_{c_1,x_1}$, obtenido a partir de aplicarle la transformación E al vector $\vec{v}_{c_0,x_0}$ (el vector "predicho").

También se verifica que el punto queda triangulado delante de ambas cámaras y que la reproyección del punto p en el plano de imagen satisface un test de $\tilde{\chi}^2$ con 2 grados de libertad.

2.5. *Simultaneous Location and Mapping*

SLAM (*Simultaneous Location and Mapping*) es un problema computacional que consiste en la construcción y actualización de un mapa en un entorno desconocido mientras simultáneamente se actualiza la posición de la cámara dentro de él. Existen diversas aproximaciones para este problema, en este proyecto se ha tratado el SLAM Visual, donde la única información usada para esta tarea son las imágenes de entrada. Por último, el escenario no se representa con su reconstrucción tridimensional, sino mediante un grafo de localizaciones conectadas entre sí, es decir, SLAM topológico basado en apariencia, cuyo sistema pionero fue FAB-MAP [19].

En este SLAM no se usa ningún tipo de información métrica, asignando cada nueva observación a una localización ya visitada, o a una nueva. En el colon, la topología del grafo es lineal, simplificando el problema (ver Figura 2.7).

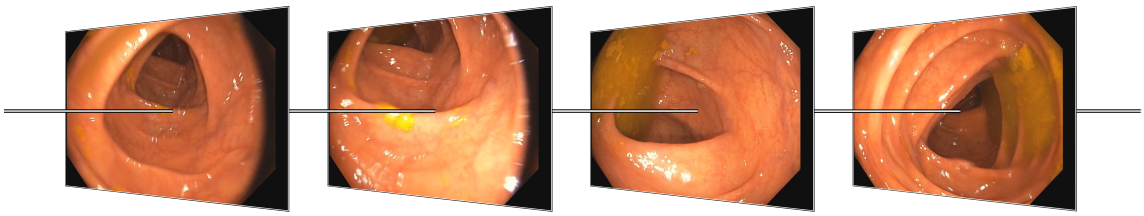


Figura 2.7: Ejemplo de mapa topológico del colon. Las aristas conectan las distintas imágenes del mapa, cuya topología es lineal y cada imagen únicamente está vinculada a la anterior y la posterior.

2.6. Métricas de evaluación

La Tabla 2.1 muestra la nomenclatura para los distintos casos de un problema de clasificación dadas 2 imágenes, siendo "Positivo" que pertenecen a la misma localización y "Negativo" que no.

		Predicho	
		Positivo	Negativo
Real	Positivo	Verdadero positivo (TP)	Falso negativo (FN)
	Negativo	Falso positivo (FP)	Verdadero negativo (TN)

Tabla 2.1: Definición de la matriz de confusión.

Para el problema de consulta a una base de datos, las definiciones son las siguientes:

- TP: Imagen emparejada correctamente.
- FP: Imagen emparejada incorrectamente.
- FN: Imagen emparejable pero no emparejada.
- TN: Imagen ocluida o borrosa.

En las pruebas, se considerará que una imagen se empareja correctamente si están tomadas a menos de t segundos.

Las métricas *accuracy*, *precision* y *recall* son muy comunes en problemas de clasificación, expresadas en su término en inglés y definidas de la siguiente forma:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.12}$$

$$precision = \frac{TP}{TP + FP} \tag{2.13}$$

$$recall = \frac{TP}{TP + FN} \tag{2.14}$$

donde *accuracy* es la tasa de aciertos del sistema, *precision* representa el porcentaje de veces que el sistema ha dado respuesta positiva correctamente y *recall* el número de todas las respuestas positivas respecto a las que podría haber dado.

Capítulo 3

Procesamiento de imágenes

Los parámetros por defecto de AKAZE están pensados para el mundo exterior, donde los contornos son más pronunciados y las esquinas están bien definidas. En este capítulo se va a estudiar el procesado de imagen necesario para facilitar la extracción de puntos de AKAZE, el ajuste de los parámetros que ofrece en la detección y descripción de puntos de interés y la eliminación de los brillos especulares. Por último, se detallará el emparejado de descriptores y la decisión de si 2 imágenes pertenecen al mismo lugar buscando el mejor *recall* pero manteniendo la *precision* al 100 %, puesto que en sistemas de SLAM es muy importante que este emparejamiento no tenga falsos positivos.

3.1. Procesado de imagen completo

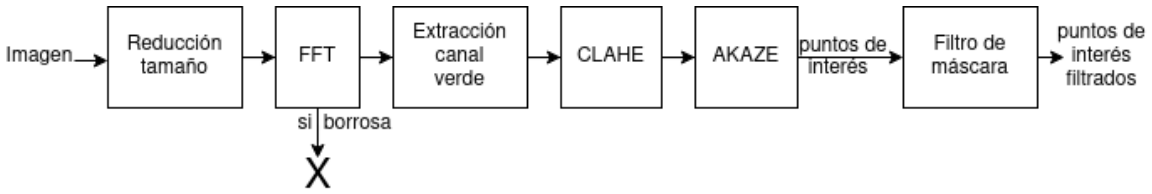


Figura 3.1: Procesado de imagen para extracción y filtro de puntos de interés.

El procesado de imagen completo es el descrito en la Figura 3.1. En primer lugar, la imagen se convierte a escala de grises mediante el uso del canal verde, se aplica CLAHE para mejorar el contraste de la imagen, se extraen puntos de interés y se aplica la máscara de brillos para evitarlos. A continuación se explican los detalles de cada procesamiento.

3.2. Tamaño, canal y aumento de contraste

En primer lugar, el tamaño de la imagen se ha reducido a la mitad usando interpolación lineal, mejorando el procesado por la probable pérdida de resolución que

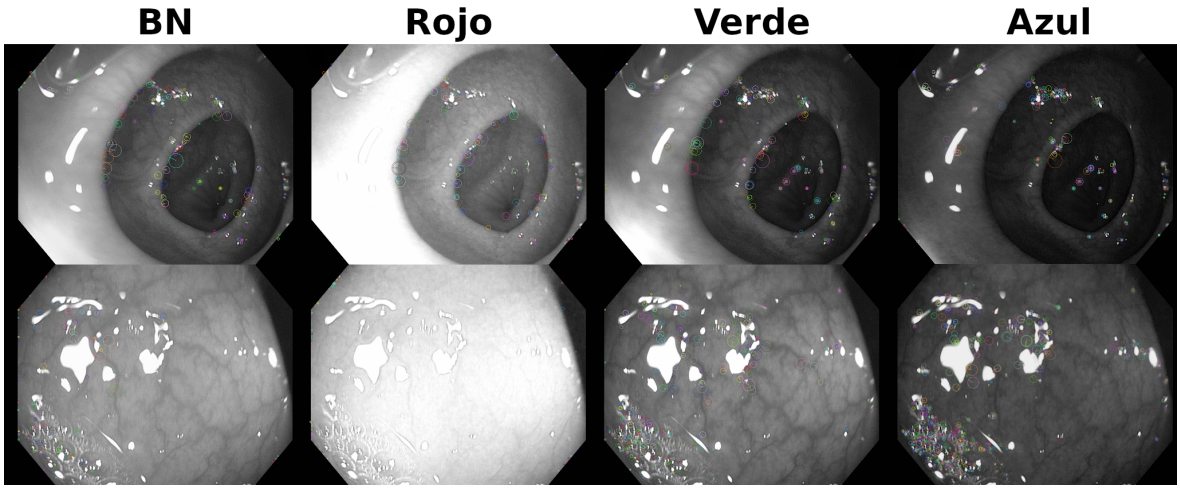


Figura 3.2: Uso de los distintos canales para la obtención de una imagen en escala de grises.

implica el patrón de Bayer de las imágenes en color originales [8].

Las imágenes tienen que estar en escala de grises para poder ser procesadas. Las opciones de transformación son dos: el paso a escala de grises mediante la ponderación de los 3 canales RGB de la imagen (opción común) o el uso directo de uno de estos canales. En la Figura 3.2 se puede observar una comparativa de estos métodos, donde se puede observar como el uso del canal verde de la imagen produce el mejor resultado, aunque el paso a escala de grises normal también funciona bien mientras que los canales R y B tienen problemas debido a los tonos de las imágenes del colon.

Sin embargo, esto no es suficiente para lograr un buen contraste en la imagen, por lo que se ha usado un algoritmo de aumento de contraste, CLAHE [20] (*Contrast Limited Adaptive Histogram Equalization*), que tras dividir la imagen en regiones ecualiza el histograma de cada una de estas, aumentando el contraste localmente. En la Figura 3.3 se puede observar el efecto de aplicarlo a las imágenes con $[8 \times 8]$ regiones y 2,0 de *clipLimit* (evita la sobre-amplificación de ruido).

3.3. Filtro de imágenes borrosas

Las imágenes borrosas u ocluidas pueden generar problemas ya que son muy similares entre sí, además de ser prácticamente imposibles de emparejar con imágenes correctas del mismo lugar. Es por esto que se filtran automáticamente. Este filtro se basa en pasar la imagen de tamaño reducido al espacio de frecuencias mediante la DFT [21] (Transformada Discreta de Fourier). Resumidamente [22], las imágenes borrosas u ocluidas no tienen características detalladas en las imágenes, que son las que tienen más energía en las altas frecuencias del espectro de frecuencias de Fourier, mientras que

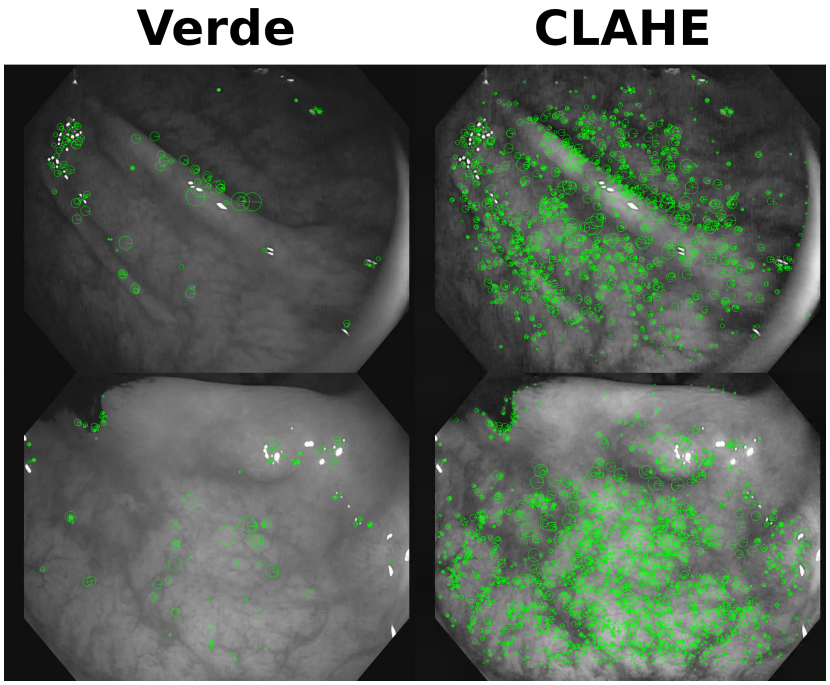


Figura 3.3: Efecto de CLAHE en imágenes del colon con [8 × 8] regiones y 2,0 de *clipLimit* en la extracción de puntos característicos.

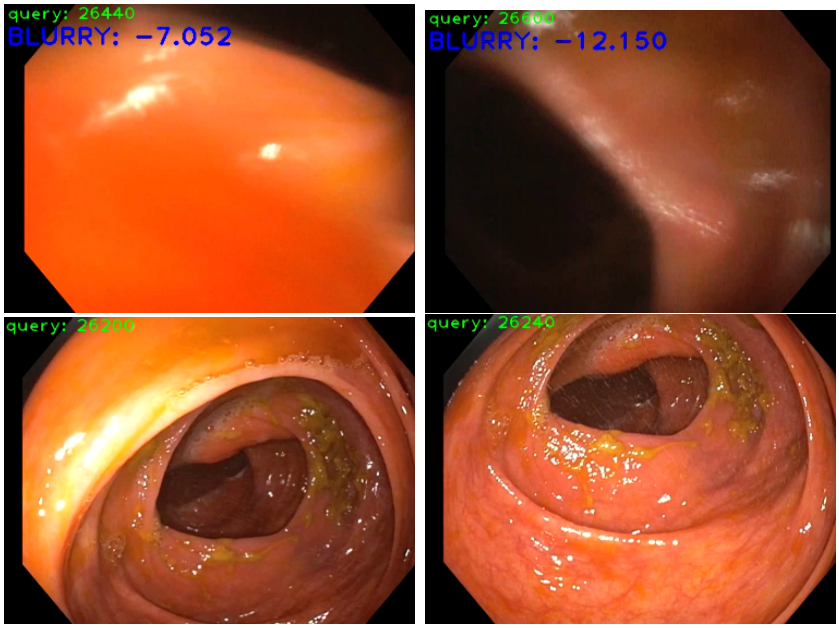


Figura 3.4: Ejemplos de imágenes detectadas como borrosas u ocluidas con su valor devuelto e imágenes no detectadas como borrosas u ocluidas.

las imágenes borrosas tienen los niveles de energía en bajas frecuencias. Tras eliminar las frecuencias bajas, se mide la energía de las frecuencias altas y cuanto más grande es el valor la imagen está más definida. Mediante una calibración previa para imágenes de endoscopio se puede establecer un umbral genérico a $-2{,}25$ y aquellas cuya medida sea menor son descartadas. La Figura 3.4 muestra ejemplos de imágenes borrosas filtradas y de imágenes aceptadas.

3.4. Parámetros de AKAZE

Los parámetros por defecto del AKAZE están ajustados para las imágenes comunes del mundo exterior, por lo que es necesario resintonizarlos para lograr buenos resultados. Estos parámetros son el número de octavas y subniveles, el umbral de detección y la función de difusividad.

El número de octavas se ha establecido a 4 puesto que se ha observado que no se extraen puntos de interés en escalas mayores y el número de subniveles también a 4, ya que usar más aumenta innecesariamente el tiempo de cómputo y no mejora el rendimiento final. La función de difusividad es crucial para lograr puntos de interés a escalas grandes, puesto que modela el difuminado adaptativo de la imagen sobre la que se aplica la detección. Para esto se han probado las distintas funciones, cuyo resultado se puede observar en la Figura 2.2. Como se puede observar, las funciones *weickert* y $g1$ preservan mejor los contornos finos de la imagen, por lo que es mejor para conservar las venas y otra información útil.

La versión de AKAZE de OpenCV no permite modificar el parámetro k de contraste de difusividad, por lo que se ha usado la implementación original [10, 5]. El parámetro k por defecto se escoge como el percentil 70 del histograma de gradientes, tendiendo a difuminar más de la cuenta al estar pensado para imágenes con más contraste (mundo exterior) por lo que se ha reducido al percentil 30. En la Figura 2.3 se puede apreciar

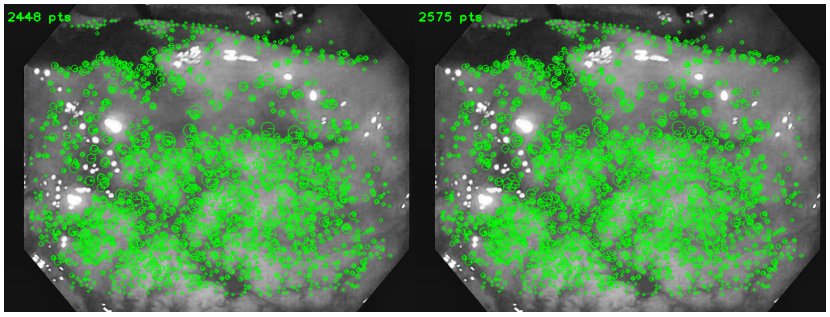


Figura 3.5: Comparación de puntos de interés antes y después del cambio al parámetro k. El número de puntos de interés aumenta ligeramente en zonas de bajo contraste, lo que mejora los resultados.

como difumina muchas menos regiones importantes y la Figura 3.5 muestra la diferencia en puntos de interés modificando este parámetro.

3.5. Máscara de brillos

Por último, es necesario definir sobre qué zonas se van a extraer puntos de interés, puesto que los reflejos especulares son zonas repetibles pero no distinguibles sobre los que hay un alto gradiente en la imagen, por lo que es necesario enmascararlo para evitar extraer puntos de interés, reduciendo el número de emparejamientos erróneos. Las zonas cuyo valor es mayor que 220 se enmascaran y la región se dilata en 10 píxeles. De esta manera se logra el resultado (ver Figura 3.6). También se enmascaran las zonas correspondientes al marco negro que rodea la región, puesto que en los bordes aparecerían puntos de interés por el mismo motivo.

3.6. Emparejamiento y verificación geométrica

El proceso de búsqueda y verificación de emparejamientos se resume en la Figura 3.7. Los descriptores de AKAZE son binarios (488 bits), por lo que la similitud de 2 descriptores se calcula como la distancia de Hamming entre estos. El emparejamiento consiste en juntar los descriptores con una menor distancia entre sí, aplicando el ratio de Lowe [23]:

$$d1 < ratio * d2 \tag{3.1}$$

para filtrar emparejamientos repetibles pero no distinguibles, puesto que si el segundo descriptor ($d2$) más cercano es similar al primero ($d1$) quiere decir que es un punto demasiado común.

Estos emparejamientos se filtran mediante la verificación de consistencia geométrica utilizando la matriz esencial (sección 2.4), para eliminar los espurios. En la Figura 3.8 se

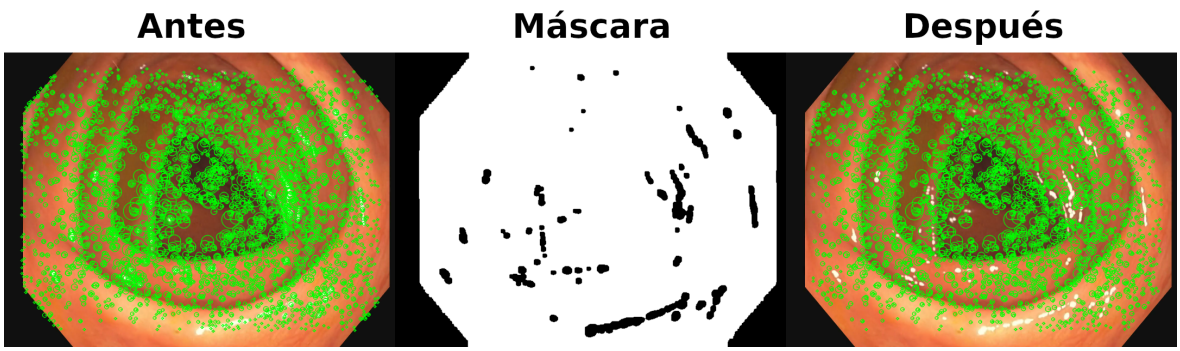


Figura 3.6: Efecto de la máscara de reflejos y borde en el filtrado de puntos de interés.

Figura 3.7: Proceso de búsqueda de emparejamientos, filtrado mediante el ratio de Lowe y la verificación de consistencia geométrica.

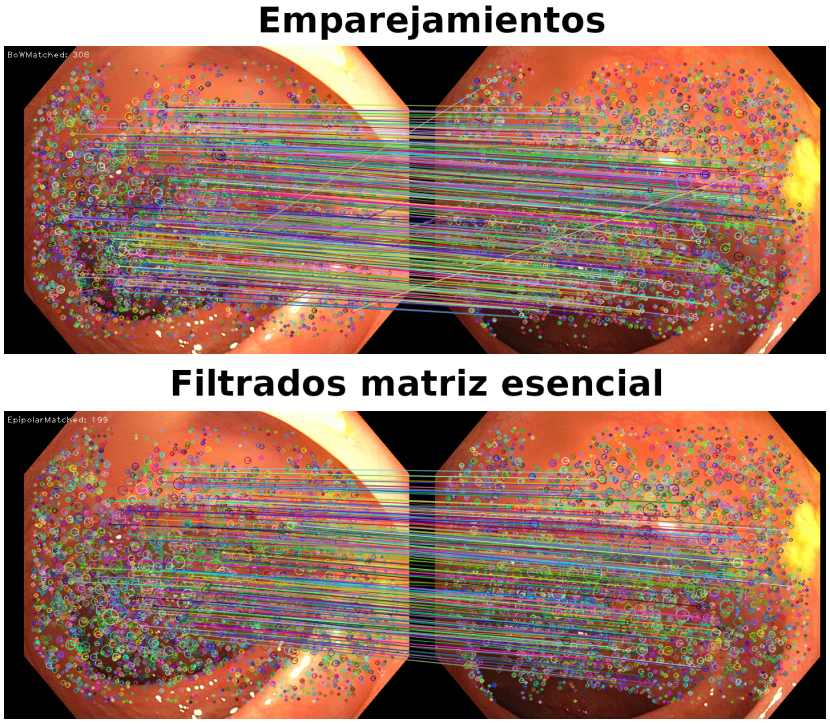


Figura 3.8: Comparación de emparejamientos antes y después de aplicar el filtrado de espurios mediante matriz esencial. Los espurios son los emparejamientos cruzados del par de imágenes superior.

puede observar un ejemplo de emparejamientos antes y después del filtrado geométrico.

3.7. Resultados

Para verificar el correcto funcionamiento de AKAZE, se han evaluado a fuerza bruta distintas combinaciones de los parámetros mencionados para observar su influencia en el emparejamiento de imágenes. Para esto, se ha usado un conjunto de datos tanto con pares positivos como con pares distractores (ver Anexo A). Las configuraciones probadas son las siguientes:

– AKAZE: Umbral a 0,00025, 4 escalas y subescalas y las funciones de difusividad *pm_g*1 y *weick*. Las funciones *pm_g*2 y *charbonnier* no se han usado porque en pruebas anteriores no daban apenas buenos resultados (ver Figura 2.2).

– Detección y descripción:

  - Blanco y negro (BN).
  - Blanco y negro con CLAHE (BNC).
  - Canal verde (V).
  - Canal verde con CLAHE (VC).
  - Canal verde aplicando CLAHE en el canal L del espacio de color LAB (VLC).
  - Detección en el canal verde tras aplicar CLAHE pero extracción en el canal verde original (VC-V).
  - Detección en el canal verde tras aplicar CLAHE al canal L del espacio de color LAB pero extracción en el canal verde original (VLC-V).

– CLAHE:

  - *clipLimit*: 1,5 y 3,0.
  - Número de regiones: $[8 \times 8]$ y $[12 \times 12]$.

– Emparejamiento:

  - Distancia de Hamming entre descriptores: 50 y 70 bits.
  - Ratio de Lowe: 0,6; 0,7; 0,8 y 0,9.

Un resumen de los resultados se puede consultar en la Tabla 3.1, donde se muestra que el mejor *recall* manteniendo la *precision* a 1,0 se basa principalmente en el uso de CLAHE con un valor alto de *clipLimit*, aumentar la distancia entre descriptores en el emparejamiento y un ratio de Lowe superior a 0,6. Sin embargo, en el resto de parámetros no es concluyente, puesto que el número de regiones y la combinación de detección-extracción (con CLAHE en detección) aparecen en los primeros puestos con *recall* similar. Por ello, se ha escogido la combinación de parámetros más simple y de menor tiempo de cómputo, que consiste en la segunda fila de la tabla.

La calibración de la matriz esencial se ha realizado manualmente, verificando los emparejamientos filtrados por esta. Por otro lado, el número mínimo de emparejamientos se establece como el valor mínimo que mantiene la *precision* al 100 % (maximizando el *recall*). Los parámetros finales son los expuestos en la Tabla 3.2.

La utilidad de la reducción del tamaño a la mitad, parámetro de difusividad de AKAZE y reducción del umbral de extracción de 0,0001 han sido probadas y sintonizadas mediante pruebas de reconocimiento de lugares en la Sección 4.3.2.

Modelo (Total=352)							*Recall*
Puesto	Detección - Descripción	CLAHE		Emparejamiento		AKAZE	
		clipLimit	Regiones	Distancia Hamming	Ratio Lowe	Difusividad	
1	VC-V	3	[12x12]	70	0.7	Weickert	**53**
2	VC	3	[8x8]	70	0.9	Weickert	50
3	VLC-V	3	[12x12]	70	0.9	Weickert	50
4	VLC-V	3	[12x12]	70	0.8	pm_g1	49
5	BNC	3	[8x8]	70	0.8	Weickert	49
6	VLC	3	[8x8]	70	0.8	Weickert	49
7	BNC	3	[12x12]	70	0.8	pm_g1	48
8	VLC	3	[8x8]	70	0.9	pm_g1	48
...							
349	V	-	-	50	0.6	pm_g1	1
350	BN	-	-	50	0.6	Weickert	1
351	V	-	-	50	0.6	Weickert	1
352	V	-	-	50	0.9	Weickert	1

Tabla 3.1: Mejores y peores resultados de las pruebas de distintos parámetros de detección y extracción, CLAHE, emparejamiento y difusividad para emparejamiento de imágenes. Ordenados de mejor a peor resultado obtenido.

Parámetro	Valor
Tasa de *inliers*	0.6
Probabilidad de éxito	0.95
Emparejamientos mínimos	20
Máximo error en píxeles de un emparejamiento	10
α para test de reproyección χ^2_α	0.01

Tabla 3.2: Parámetros del RANSAC y matriz esencial para filtrado de espurios.

Capítulo 4

Reconocimiento de lugares en endoscopias

El reconocimiento de lugares durante una colonoscopia es una parte muy importante debido a la cantidad de oclusiones que el endoscopio sufre durante una intervención. En este apartado se va a detallar el procedimiento seguido y los resultados obtenidos en distintas secuencias y experimentos.

4.1. Creación del vocabulario

El primer paso para el reconocimiento de lugares basado en bolsas de palabras es generar un vocabulario de palabras visuales. Este vocabulario es el que permite transformar un conjunto de descriptores en una bolsa de palabras. Por ello, cuanto mejor represente el vocabulario al problema, las bolsas de palabras contendrán información más representativa y repetible de la imagen.

Siguiendo lo indicado en [9], se han escogido unos parámetros de tamaño del árbol de $k_w = 10$ (parámetro de ramificación) y $L_w = 6$ (niveles del árbol), dando lugar a un total de 10^6 hojas teóricas. Este valor es lo suficientemente grande como para representar todo el espacio de descriptores y correcto para no provocar problemas por hojas que cubran muy poco rango.

El vocabulario ha sido generado mediante únicamente imágenes de colonoscopias (secuencias 27 y 35 del Dataset público [11] y 327 del Dataset I), dado un conjunto de $8{,}5 * 10^6$ descriptores, que dan lugar a un árbol con $9{,}9 * 10^5$ hojas (palabras visuales).

4.2. Procedimiento de reconocimiento de lugares

El procedimiento de reconocimiento de lugares es el descrito en el algoritmo 1. Dados un conjunto de candidatos obtenido de la base de datos (DBoW, *Database Bag of Words*) mediante la comparación de bolsas de palabras, para cada uno de ellos se

trata de obtener un conjunto de emparejamientos robustos y si este es superior a 20 se considera que las imágenes corresponden a una misma localización. El uso de la verificación de consistencia geométrica mediante la matriz esencial es necesario porque la bolsa de palabras puede dar resultados erróneos, por lo que sirve para validar los candidatos obtenidos de esta.

Algorithm 1 Reconocimiento de lugares (nuevaImagen)

```
1: candidatos ← DBoW.consulta(nuevaImagen)
2: for all candidato in candidatos do
3:     emparejamientos ← emparejarDescriptores(nuevaImagen, candidato)
4:     emparejamientosFiltrados ← matrizEsencial(emparejamientos)
5:     if |emparejamientosFiltrados| > minEmparejamientos then
6:         return candidato
7: return No hay candidato
```

El número de candidatos a escoger para la verificación geométrica se ha fijado a 3 debido a que la aportación en el *recall* del resto de candidatos no es lo suficientemente significativo como para invertir el tiempo de cómputo necesario en ello. En la Figura 4.1 se puede apreciar que a partir del segundo o tercer candidato, pese a que la *accuracy* de la bolsa de palabras se mantiene alta, el *recall* aportado no es significativo.

4.3. Validación experimental

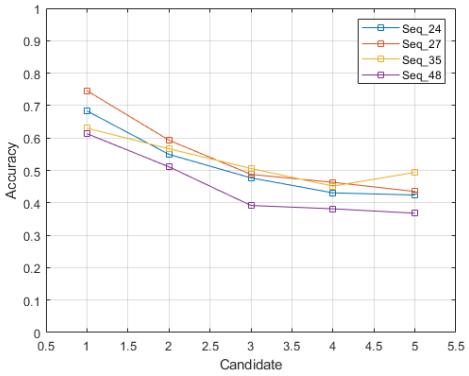
Para validar el funcionamiento de esta aproximación, se van a realizar varios experimentos, con el objetivo de observar el comportamiento para imágenes del mismo lugar muy cercanas en tiempo e imágenes más distantes, donde el colon, al ser un entorno dinámico, ha sufrido deformaciones.

En las rutinas comunes de colonoscopia, el primer paso es introducir el colonoscopio hasta llegar al ciego (parte final del colon, ver Figura 4.2). La entrada del endoscopio suele sufrir constantes oclusiones a diferencia de la salida que se realiza con más suavidad para poder observar el colon del paciente.

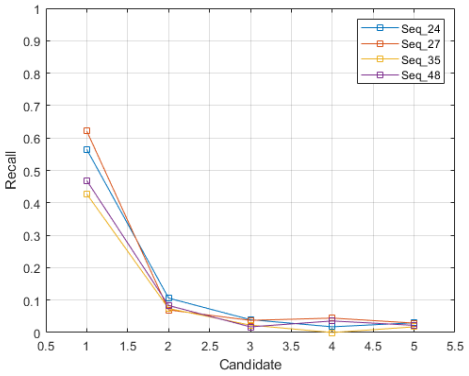
4.3.1. Metodología

La metodología de validación de esta aproximación consiste en tratar de emparejar imágenes cercanas en tiempo de una misma trayectoria de salida del endoscopio. Esto nos permite evaluar la capacidad del método para relocalizar el endoscopio tras una pérdida del sistema de SLAM por oclusiones, chorros de agua para limpieza, o movimientos bruscos.

Por ello, la prueba se define de la siguiente manera:



(a) *Accuracy* de DBoW para cada candidato.



(b) *Recall* de DBoW aplicando verificación geométrica para cada candidato.

Figura 4.1: Comparación de la *accuracy* de DBoW para cada candidato y el *recall* que aporta realmente al reconocimiento de lugares en las distintas secuencias.

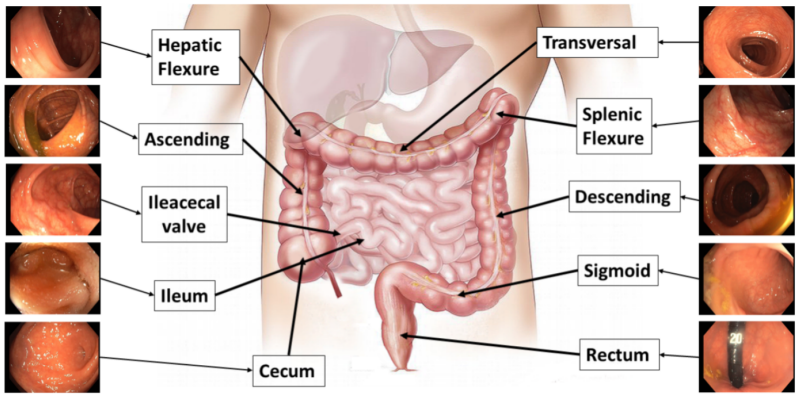


Figura 4.2: Partes del colon [11] (*cecum* = ciego).

1. Se introducen en la base de datos imágenes con un periodo Δt de la salida del endoscopio, denotando el inicio de la salida como el instante T_s. Por ello, se introducen los fotogramas F $t.q.$ $T_f \in \{T_s + i\Delta t, \forall i \in \mathbb{N}\}$.

2. Se consultan los fotogramas intermedios F $t.q.$ $T_f \in \{T_s + \frac{\Delta t}{2} + i\Delta t, \forall i \in \mathbb{N}\}$

3. Se evalúa el resultado como la distancia en tiempo entre consulta y resultado (ver sección 2.6), manteniendo la *precision* al 100 %. Es decir, que en ningún caso se va a obtener un emparejamiento erróneo. La métrica de evaluación usada va a ser el *recall* a t segundos, por lo que *Recall*@40s es el *recall* del sistema usando como umbral un máximo de 40 segundos.

En las pruebas $\Delta t = 0{,}8s$, por lo que se están tratando de emparejar imágenes a 0,4s de distancia temporal. El intervalo de 40s puede resultar muy grande, pero posteriormente se mostrará el correcto funcionamiento con umbrales menores. Este

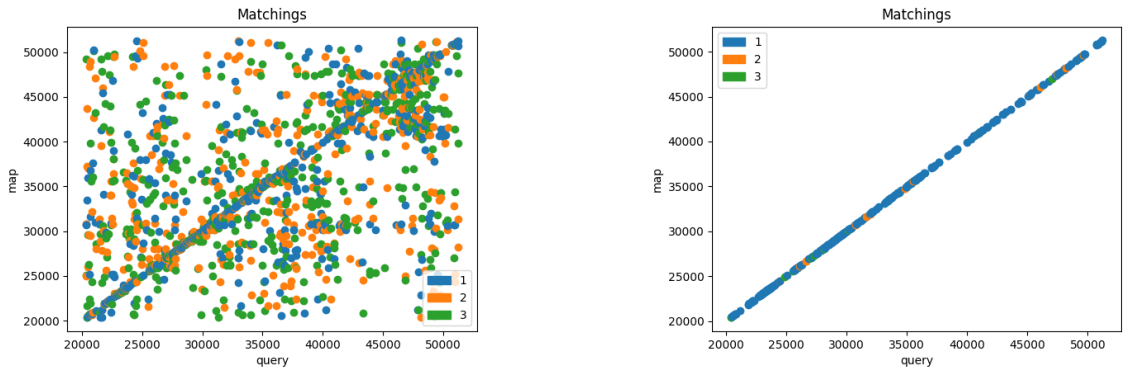
valor se debe a que, en determinadas secuencias, el endoscopio (a decisión del cirujano) se queda estático o con mínimo movimiento durante largos periodos de tiempo, por lo que el reconocimiento de lugares puede escoger como candidatos localizaciones separadas esta cantidad de tiempo y la métrica no lo detectaría.

4.3.2. Resultados

En primer lugar, en la Figura 4.3 se puede observar la posición de la imagen consulta, los candidatos devueltos por la base de datos (DBoW) y los que han superado la verificación geométrica. Se puede apreciar que este método filtra correctamente los candidatos erróneos al quedar la figura en diagonal (consulta y candidato muy cercanos en tiempo).

La Tabla 4.1 muestra el *recall* obtenido en distintas secuencias conforme se van incorporando las mejoras de procesamiento discutidas anteriormente. En esta tabla el umbral es de $40s$ porque en la secuencia 24 hay casos en los que se queda en la misma zona un largo periodo de tiempo. La configuración que obtiene la mejor media de resultados corresponde a reducir el tamaño a la mitad, cambiar en el umbral de extracción de 0,00025 a 0,0001 y disminuir del parámetro k de difusividad a 0,3. El umbral adaptativo no se muestra útil respecto a la configuración anterior, lo que puede ser causado porque los descriptores de los nuevos puntos de interés añaden ruido a las bolsas de palabras.

Los resultados finales se pueden consultar en la Tabla 4.2. Excluyendo oclusiones, la base de datos de bolsa de palabras llega a reconocer el lugar un $57 \sim 78\,\%$ de las veces sin ningún falso positivo (manteniendo la *precision* al $100\,\%$) y se puede apreciar que a partir de los $8s$ apenas varía en las secuencias.



(a) Tres primeros candidatos devueltos por DBoW para cada consulta.

(b) Primer candidato que supera la verificación geométrica (si hay).

Figura 4.3: Emparejamientos de fotogramas de la salida de la secuencia 48.

Recall@40s								
Secuencia	Original	Reducción tamaño	Umbral extracción	Difusividad (k)				Umbral adaptativo
				0.1	**0.3**	0.5	0.7	
Seq_24	59.53	59.67	71.11	68.74	67.88	68.18	71.11	68.11
Seq_27	67.59	70.91	74.02	73.6	77.27	70.35	74.02	77.78
Seq_35	47.77	49.19	54.76	52.68	57.18	54.07	54.76	54.84
Seq_48	48.48	54.05	57.91	58.34	60.32	57.85	57.91	59.86
Media	55.83	58.46	64.45	63.64	**65.66**	62.61	64.45	65.15

Tabla 4.1: *Recall*@40s añadiendo las distintas mejoras propuestas a la sintonización de AKAZE para reconocimiento de lugares. Cada columna muestra el *recall* resultante de añadir a la configuración de la columna anterior la columna actual.

Recall					
	2s	4s	8s	20s	40s
Seq_24	63.93	65.77	68.05	69.66	70.62
Seq_27	75.52	77.54	78.48	78.67	78.67
Seq_35	54.07	55.75	56.59	56.59	56.59
Seq_48	57.44	57.44	57.44	57.44	57.44

Tabla 4.2: *Recalls* del reconocimiento de lugares mediante bolsa de palabras manteniendo la *precision* al 100,00 %, con distintas distancias en tiempo para verificar el correcto funcionamiento. Las variaciones en *recall* de los resultados con *Recall*@40s varían ligeramente respecto a los de la Tabla 4.1 porque RANSAC es un método probabilista.

4.4. Tiempo de cómputo y aceleración mediante GPU

La Tabla 4.3 y la Figura 4.4 muestran los tiempos de ejecución de las distintas partes del sistema, en el caso de AKAZE y del emparejamiento, con sus versiones en CPU y GPU.

El proceso de extracción de puntos de interés en AKAZE (que incluye CLAHE) en CPU es muy costoso y presenta medidas anómalas que parecen indicar algún error de implementación de la librería, puesto que no se ha observado correlación entre los espurios y el número de puntos de interés extraídos. El uso de la GPU en CLAHE mediante su implementación en CUDA de OpenCV y la librería de AKAZE en GPU [12] reducen mucho el tiempo de cómputo, siendo mucho más viable en tiempo real. El espurio de la GPU se debe a la primera ejecución del algoritmo por la preparación de esta.

El RANSAC suele tardar entre $10 \sim 50ms$ pero con una gran cantidad de casos en los que se aleja del rango intercuartílico. Esto se debe a que el número de iteraciones es adaptativo al porcentaje de emparejamientos correctos encontrados, lo que provoca que si no encuentra un buen conjunto realiza el máximo número de iteraciones. Este

Tiempo (ms)	Media	Mediana
akazeCPU	83.96	60.37
akazeGPU	13.95	11.11
queryDBoW	10.84	10.38
matchCPU	17.19	14.30
matchGPU	3.32	2.87
RANSAC	46.04	11.90

Tabla 4.3: Tiempo de cómputo de las distintas partes del sistema.

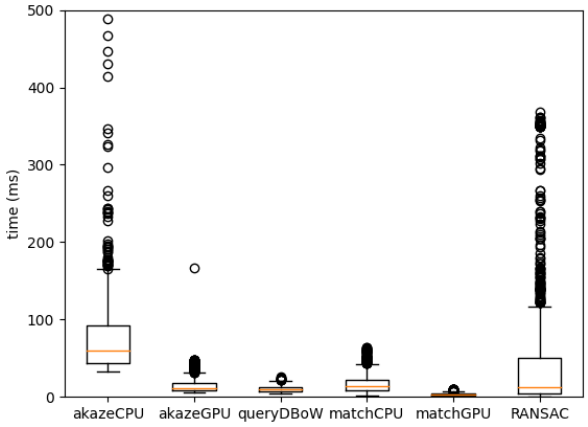


Figura 4.4: Diagramas de caja del tiempo de cómputo (eje y limitado a $500ms$).

es principalmente el cuello de botella del sistema, puesto que no se dispone de una implementación en GPU que permita acelerar el cálculo. Tampoco se pueden aplicar directamente otros algoritmos más eficientes como el MAGSAC++ [24, 25], puesto que las implementaciones asumen un modelo de cámara oscura (*pinhole*) frente al modelo gran angular que tiene el endoscopio.

Cabe destacar que en una consulta solo se realiza una vez la extracción de puntos de interés y la consulta, mientras que los emparejamientos y el RANSAC se realizan por cada candidato obtenido de DBoW. Dado el alto coste por candidato, el mejor número de candidatos estaría entre 2 y 3.

4.5. Conclusiones

El método de reconocimiento de lugares usando AKAZE como herramienta de detección y descripción de puntos de interés, junto al procesado de imagen realizado, obtienen muy buen *recall* para problemas de relocalización tras una oclusión y el uso de la GPU supone una gran mejora en el tiempo de ejecución para tiempo real. Sin embargo, no se han conseguido resultados positivos en los dos problemas más difíciles: reconocer a la salida del endoscopio lugares que se vieron a la entrada y reconocer lugares que se observaron en una exploración previa del paciente, probablemente debido a las deformaciones que sufre el colon. En estos problemas, las técnicas basadas en redes neuronales parecen más prometedoras.

Capítulo 5

SLAM topológico basado en apariencia

En este capítulo se va a presentar el uso de un filtro de Bayes discreto [26] para localizar el endoscopio, con el objetivo de filtrar espurios y lograr un movimiento suave de la localización a lo largo del mapa, priorizando la vecindad. Tras ello, se construirá un sistema de SLAM topológico basado en apariencia capaz de crear grafos con topologías más complejas, no necesariamente lineales.

5.1. Localización probabilista

En el capítulo anterior el sistema obtenía un resultado basado únicamente en las observaciones vistas en el instante actual. En este modelo, se hace uso de una estimación de probabilidad para cada instante t, teniendo en cuenta el posible movimiento de la cámara, su distribución de probabilidad en $t-1$ y usando las observaciones del instante actual. El método se basa en la regla de Bayes:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} = \frac{P(B \mid A)P(A)}{\sum_{i=1}^{n} P(B \mid A_i)P(A_i)} = \eta P(B \mid A)P(A) \tag{5.1}$$

siendo η el factor de normalización. En nuestro caso, se puede estimar la probabilidad $p(x_t \mid z_{1:t}, u_{1:t})$, es decir, la posición del endoscopio x en el instante t conociendo las observaciones (mediciones de sensores) z de todos los instantes anteriores y todas las acciones de control u aplicadas (movimiento del endoscopio):

$$p(x_t \mid z_{1:t}, u_{1:t}) = \eta p(z_t|x_t)p(x_t \mid z_{1:t-1}, u_{1:t}) \tag{5.2}$$

Conociendo la posición anterior del endoscopio x_{t-1}, $p(x_t)$ se puede obtener mediante $p(x_t \mid x_{t-1}, z_t, u_t)$, es decir, no es necesario conocer ni las observaciones ni las acciones de control anteriores al instante t, tal que:

$$p(x_t) = \eta p(z_t \mid x_t)p(x_t \mid x_{t-1}, u_t) \tag{5.3}$$

Por tanto, el algoritmo que resuelve la estimación de la probabilidad $p(x_t)$ en el instante t para toda localización k es el siguiente:

Algorithm 2 Filtro de Bayes discreto [26]

1: **for** all k **do**
2: $\quad \overline{p}_{k,t} = \sum_i p(X_t = x_k \mid u_t, X_{t-1} = x_i) p_{i,t-1}$ ▷ Predicción
3: $\quad p_{k,t} = \eta p(z_t \mid X_t = x_k) \overline{p}_{k,t}$ ▷ Actualización

El paso 2 del algoritmo 2 representa la predicción de localización a partir del conocimiento de la localización anterior y las acciones de control tomadas y el paso 3 corresponde a la actualización de las probabilidades $p(x_t)$ en base a las observaciones.

5.2. Adaptación del filtro Bayesiano

Para la predicción, la variable u_t (control del endoscopio) es desconocida puesto que el movimiento es decidido por el cirujano, por lo que se establece $1/(n+1)$ como probabilidad de avanzar, quedarse en el sitio e ir hacia atrás, siendo n el número total de localizaciones vecinas a distancia $d \leq 2$.

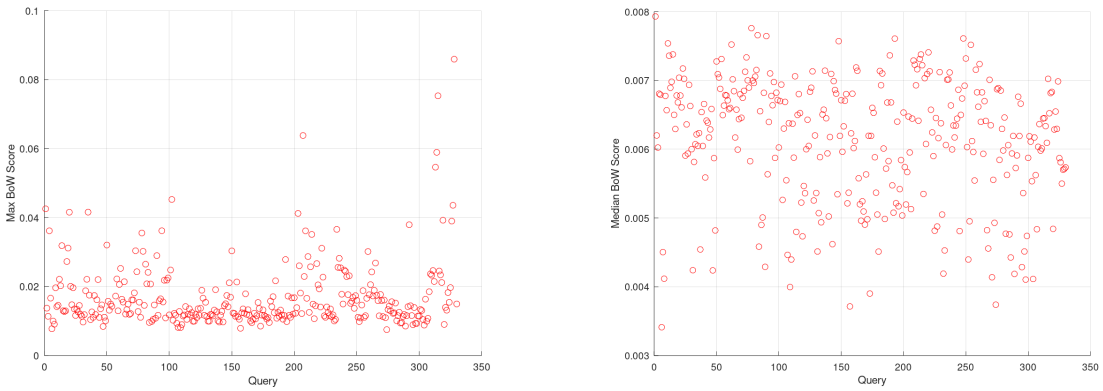
Para la actualización, es necesario estimar $p(z_t|X_t = x_k)$, la verosimilitud de observar z_t desde x_k en el instante t. Para ello, tomamos:

$$p(z_t|X_t = x_k) = \rho \; s(v_t, v_{x_k}) \tag{5.4}$$

donde s es la medida de similitud entre las bolsas de palabras de la imagen tomada en el instante t y la bolsa de palabras de la localización x_k y ρ es el *bonus* por consistencia geométrica, que vale 1 si no se supera la consistencia y 3 si se supera. La verosimilitud está limitada a 0,99 si hay *bonus* y a 0,85 si no.

La Figura 5.1 muestra la distribución de similitudes obtenidas por el mejor candidato para cada imagen de consulta y la mediana de similitudes con el resto de imágenes del mapa. Para que el algoritmo funcione en tiempo real, tanto el cálculo de similitud entre bolsas de palabras como la verificación de la consistencia geométrica solo se pueden calcular para unos pocos candidatos, que son los que devuelve la bolsa de palabras. Al resto de lugares del mapa se les asigna una verosimilitud constante de 0,004, algo inferior a la mediana observada experimentalmente.

Si la imagen de entrada es borrosa, únicamente se aplica el paso de predicción del filtro de Bayes, ya que la actualización únicamente introduce ruido en la localización. En la Figura 5.2 se puede ver como tras varias imágenes detectadas como borrosas, la distribución de probabilidad se asemeja a una Gaussiana.



(a) Similitud del mejor candidato.

(b) Mediana de las similitudes de todas las imágenes de mapa.

Figura 5.1: Valores de similitud obtenidas para cada imagen de consulta.

5.3. Construcción automática del mapa

No todos los fotogramas del vídeo aportan información relevante para la construcción de un mapa. Por ello, se ha realizado una selección de fotogramas clave. Pese a que la topología del colon es lineal, un mapa topológico tiene que tener en cuenta que varias imágenes tomadas en la misma localización pueden representar paredes laterales diferentes del colon (rotación de la cámara), por lo que se han permitido bifurcaciones en el grafo.

Para lograrlo, se ha realizado un sistema de SLAM usando la localización Bayesiana y varias heurísticas para la adición de fotogramas clave. El proceso se puede observar en el algoritmo 3, donde tras realizar la localización Bayesiana, se comprueba la similitud mediante bolsa de palabras entre la nueva imagen y la localización actual. Si baja de un determinado umbral o hace más de $6s$ de la última actualización se añade esta imagen como una localización vecina de la actual con probabilidad 0,9, y el resto se actualizan en concordancia.

Algorithm 3 SLAM topológico (z)

1: $locActual \leftarrow$ **LocalizaciónBayesiana**(z) ▷ z es la nueva observación
2: **if** $s(v_z, v_{locActual}) < umbralBoW \vee 6s$ **then**
3: **for** all i **do** $localizaciones_i.p \leftarrow localizaciones_i.p * 0{,}1$ ▷ p es la probabilidad
4: $nuevaLoc = loc(z)$
5: $nuevaLoc.p \leftarrow 0{,}9$
6: **actualizarVecinos**($locActual, nuevaLoc$)
7: $localizaciones \leftarrow localizaciones \cup \{nuevaLoc\}$

La construcción simultánea del mapa y localización del endoscopio altera el paso de predicción de la localización (ver algoritmo 2), puesto que al estar explorando un

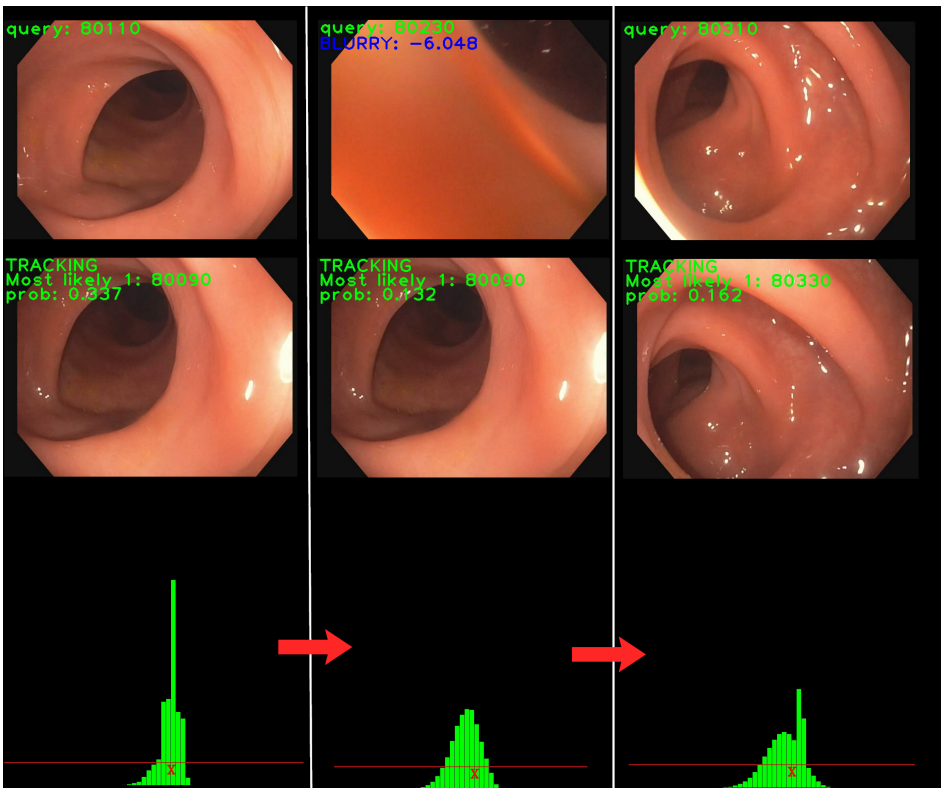


Figura 5.2: Evolución de $p_{k,t}$ y el estado del sistema tras una oclusión y la relocalización posterior.

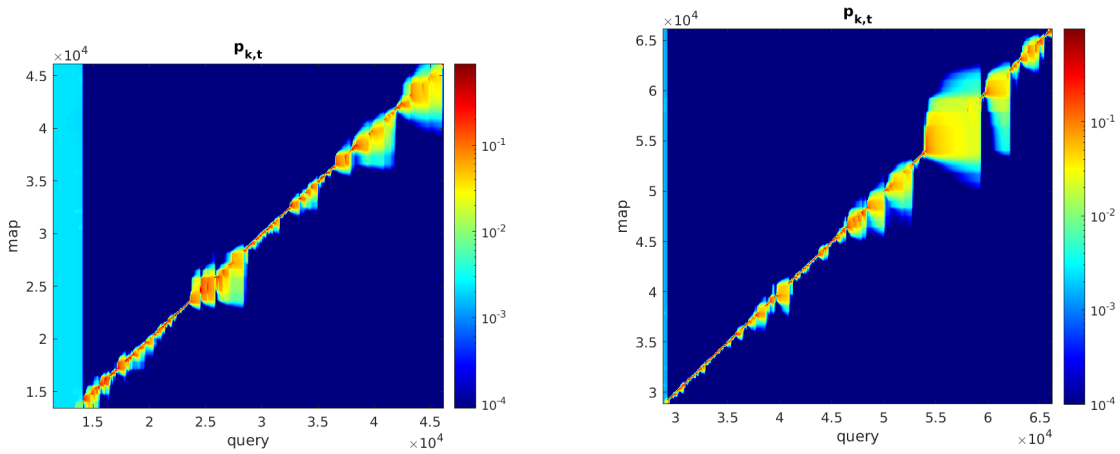
lugar que puede ser desconocido la probabilidad no solo se reparte entre el mismo nodo y los vecinos a distancia $d \leq 2$, sino también a una posible localización desconocida adyacente. Por tanto, cada nodo reparte $\frac{1}{n+2}$ a los vecinos y se queda $\frac{2}{n+2}$.

5.4. Validación experimental

5.4.1. Localización Bayesiana con mapa previo

Las distribuciones de probabilidad obtenidas para las distintas secuencias se pueden observar en la Figura 5.3. Como se puede apreciar, la distribución en forma de diagonal de probabilidad muestra el buen funcionamiento de la localización. En algunas zonas la distribución se expande en lugar de dar picos de probabilidad debido a la aparición de imágenes borrosas que el filtro descarta, donde únicamente se realiza la predicción y no la actualización del algoritmo 2.

En determinadas aplicaciones es necesario establecer un umbral de probabilidad para separar los casos en los que la localización es desconocida y los que no. Este valor es de 0,038, obtenido visualizando las gráficas *precision-recall* (ver Figura 5.4) de las distintas secuencias y escogiendo un valor que mantiene la *precision* lo más alto posible pero sin descuidar el *recall*. Manualmente se puede verificar que la *precision* es mayor



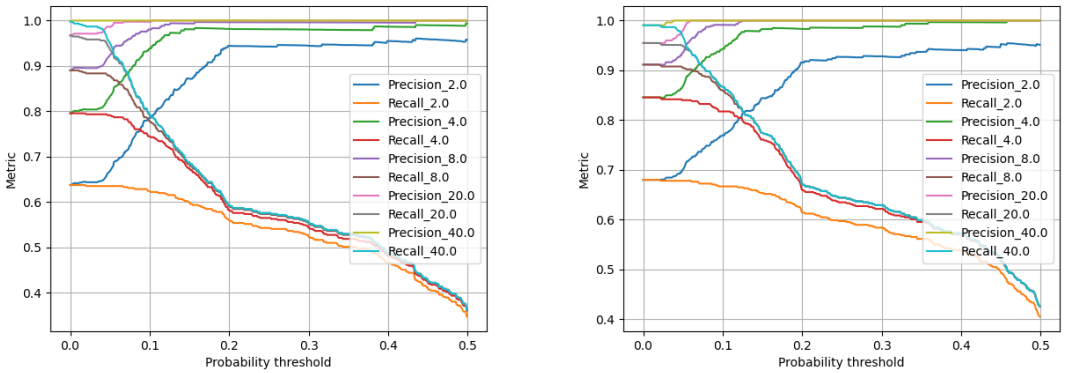
(a) Secuencia 24

(b) Secuencia 27

(c) Secuencia 35

(d) Secuencia 48

Figura 5.3: Probabilidad obtenida por el filtro Bayesiano de estar en los distintos lugares del mapa para cada instante de consulta. El sistema arranca con distribución equiprobable (zona azul claro) y conforme empiezan a emparejarse imágenes, la probabilidad se concentra en la zona correcta.



(a) Secuencia 24

(b) Secuencia 27

Figura 5.4: Gráficas *precision-recall* para varias secuencias y distintos umbrales de tiempo en segundos.

a la que aparece en la gráfica, pero en esta sección se asume un cierto error. Este valor se usará durante todas las pruebas.

En la Tabla 5.1 se pueden observar los resultados obtenidos para distintas configuraciones de número de candidatos de la bolsa de palabras con los que se calcula una verosimilitud real y la consistencia geométrica. Como se puede apreciar, los mejores resultados se obtienen con la mayor bonificación de consistencia geométrica y número de candidatos. La mejora al aumentar la bonificación por consistencia geométrica es mucho mayor que con el número de candidatos y sin esta la métrica cae considerablemente. Cuantos más candidatos, más consistencias geométricas se tienen que realizar, que es muy costoso (por el RANSAC) para lograr una mejora de apenas 1 % de *recall*. Por ello, la configuración elegida es $N = 2$ y $\rho = 3{,}0$.

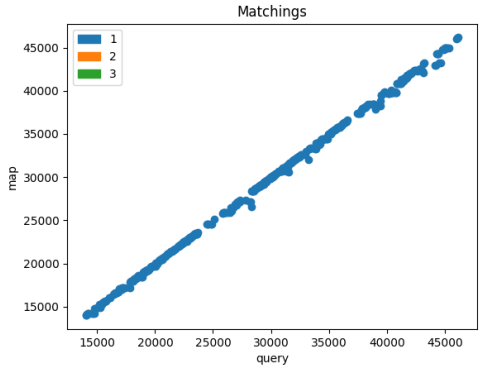
Parámetros		*Recall*@40s			
ρ	Nº DBoW	Seq_24	Seq_27	Seq_35	Seq_48
1	2	77.58	93.78	63.94	59.80
1	3	82.24	95.86	70.00	65.25
1	5	87.25	96.42	76.06	72.57
2	2	94.94	98.31	93.33	96.59
2	3	96.48	98.49	94.24	97.27
2	5	97.58	98.49	96.06	96.59
3	**2**	98.24	98.87	97.57	97.78
3	3	**98.68**	98.87	95.76	98.98
3	5	**98.68**	**99.44**	97.27	**99.32**
3	10	**98.68**	**99.62**	**99.70**	**99.32**

Tabla 5.1: Pruebas del sistema usando la métrica *Recall*@40s para los distintos valores de bonus de consistencia geométrica y número de candidatos de DBoW.

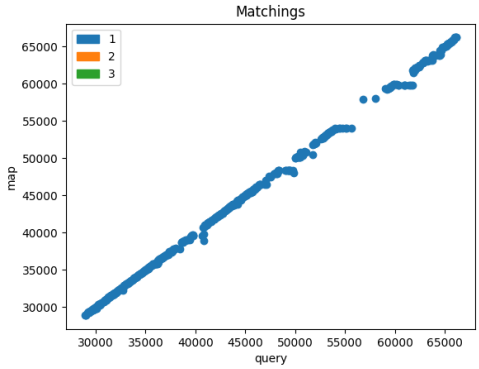
Las métricas de la configuración final del sistema se pueden consultar en la Tabla 5.2 y la Figura 5.5 muestra el primer candidato para cada consulta en las distintas secuencias. En comparación con el reconocimiento de lugares, las métricas caen mucho cuanto más pequeño es el umbral. Esto se debe a que en vez de dar una localización exacta, trata de hacer movimientos suaves por el mapa al expandir la probabilidad solo entre las localizaciones vecinas.

Secuencia	*Recall*				
	2s	4s	8s	20s	40s
Seq_24	70.55	79.34	88.35	95.60	98.24
Seq_27	77.40	84.18	91.15	95.10	98.87
Seq_35	56.06	67.27	78.78	90.91	97.58
Seq_48	66.95	76.83	87.22	94.72	97.79

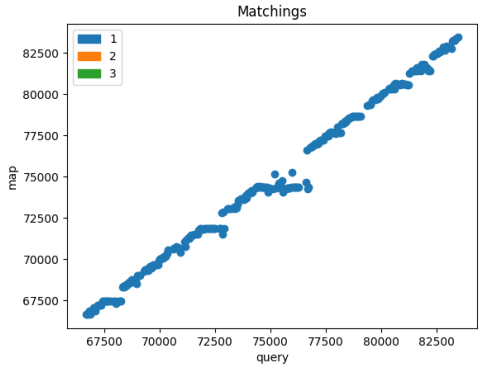
Tabla 5.2: *Recalls* de la localización Bayesiana con distintos valores de distancia temporal para verificar el correcto funcionamiento.



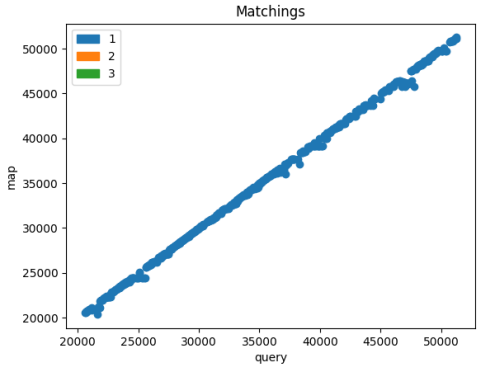
(a) Secuencia 24



(b) Secuencia 27



(c) Secuencia 35



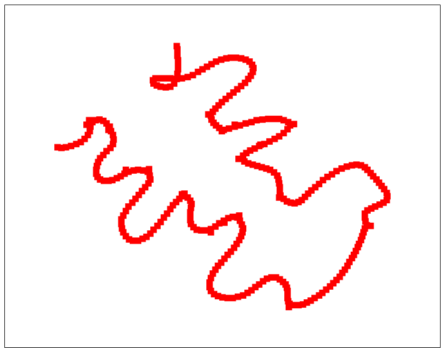
(d) Secuencia 48

Figura 5.5: Candidato más probable que supera el umbral de probabilidad. Las zonas vacías se deben principalmente a oclusiones o imágenes borrosas donde la mayor probabilidad no ha superado el umbral.

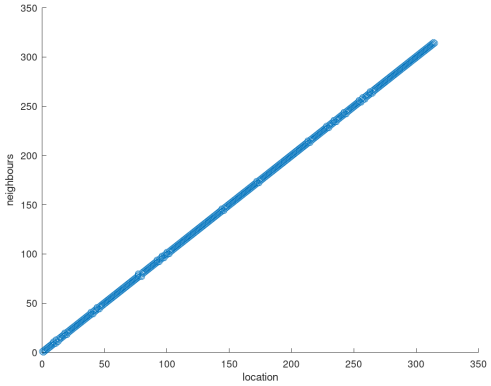
5.4.2. SLAM topológico basado en localización Bayesiana

Al terminar el SLAM, la topología de los mapas construidos es principalmente lineal (ver Figura 5.6). En algunos casos aparecen bifurcaciones debido a la observación de las paredes, cuando el movimiento no sigue la topología del colon o imágenes parecidas a fotogramas clave pero que se han decidido añadir. La Figura 5.6c ilustra un caso de bifurcación en un mapa con muchos fotogramas clave añadidos, donde distintas vistas de la misma localización han sido añadidos como ramas.

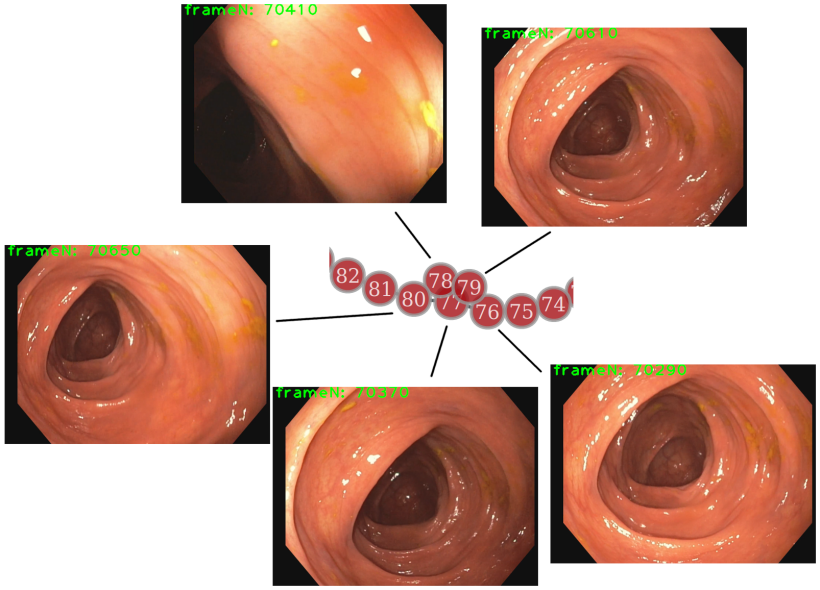
La Figura 5.7 muestra la variación en *recall* para cada secuencia dados distintos tamaños de mapa construidos. La construcción del mapa mejora el resultado en algunas secuencias como la 35, 48 y 24 aunque tiene problemas al final de la secuencia 27, disminuyendo la métrica. Las mejores métricas obtenidas corresponden a los 2 mayores mapas, con un tamaño de alrededor del 80 % del original, aunque el mapa más pequeño también obtiene muy buenas métricas con un tamaño del 58 % respecto al original. Nótese que la mejora es notable incluso en los umbrales de tiempo más restrictivos pese a que la selección de fotogramas clave limita esta métrica.



(a) Grafo del mapa topológico.



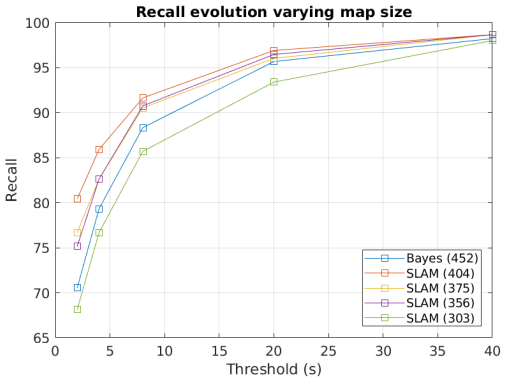
(b) Matriz de adyacencia del grafo.



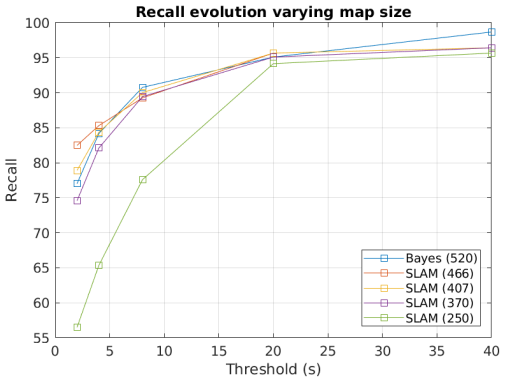
(c) Ejemplo de bifurcación: topología del grafo e imágenes de cada nodo.

Figura 5.6: Distintas visualizaciones del mapa topológico generado con la secuencia 35. La topología es principalmente lineal.

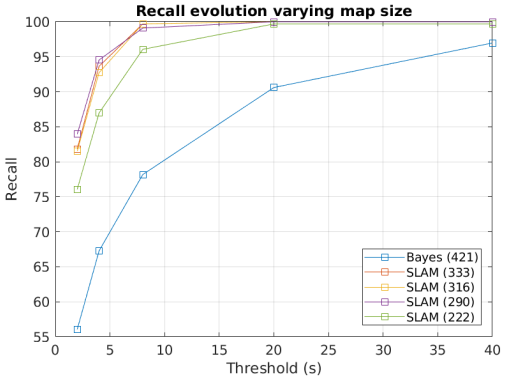
La configuración final de Bayes corresponde al umbral de similitud de bolsas de palabras a 0,017 (línea amarilla en la Figura 5.7), logrando un *recall*@8s del $90 \sim 99\,\%$ en las distintas secuencias, frente al $78 \sim 92\,\%$ de la localización Bayesiana. Esta mejora se debe a la excesiva cantidad de localizaciones del mapa anterior al no seleccionar fotogramas clave.



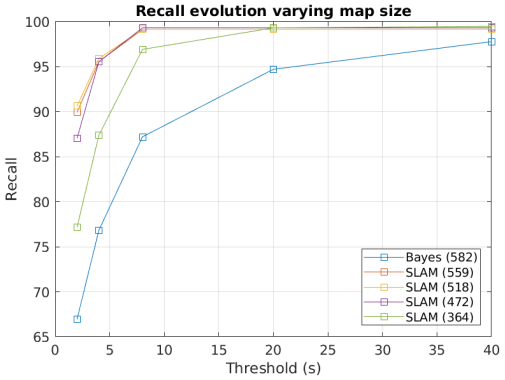
(a) Secuencia 24



(b) Secuencia 27



(c) Secuencia 35



(d) Secuencia 48

Figura 5.7: Evolución del *recall* para distintos tamaños de mapa en comparación con la localización Bayesianas. Los umbrales de similitud para añadir un fotograma clave en SLAM son: 0,026; 0,017; 0,014 y 0,010.

5.5. Análisis temporal

El uso de un mayor número de candidatos supone un aumento en el *recall* del sistema. Sin embargo, esto también conlleva un incremento en el tiempo de cómputo debido a la verificación de la consistencia geométrica. La configuración final (2 candidatos) supone un rendimiento medio usable en tiempo real, como muestra la Tabla 5.3 y la Figura 5.8, donde la mediana se encuentra en $125ms$, pero dando picos de $800ms$ debido al RANSAC. El cómputo de la fórmula de Bayes dadas las verosimilitudes es despreciable.

En el conteo total del tiempo de ejecución no se han contado los casos de solo actualización del control por imágenes borrosas u ocluidas.

Tiempo (ms)	Media	Mediana
akazeGPU	13.945	11.107
queryDBoW	10.836	10.377
matchGPU	3.320	2.875
RANSAC	46.043	11.900
Predicción	0.002	0.002
Actualización	0.002	0.002
Total	148.846	125.327

Tabla 5.3: Métricas del tiempo de cómputo de la localización Bayesiana.

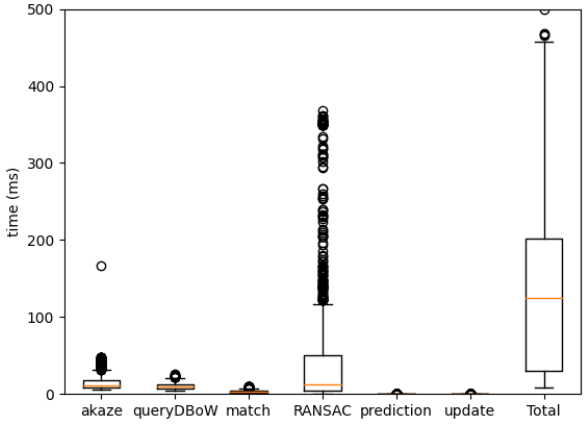


Figura 5.8: Diagramas de caja del tiempo de cómputo de la localización Bayesiana (eje y limitado a $500ms$).

Capítulo 6

Conclusiones y trabajo futuro

En este trabajo se ha demostrado el buen funcionamiento de AKAZE y bolsas de palabras visuales en tareas de reconocimiento de lugares y localización dentro de una misma salida de una colonoscopia, es decir, después de que el cirujano llega al ciego (parte final del colon) y comienza a extraerlo. Sin embargo, en otras tareas como revisita o la entrada contra la salida no funciona. Esto se puede deber a que, tras deformaciones, AKAZE no es lo suficientemente robusto y los puntos de interés no son tan repetibles, por lo que sus descriptores son distintos y la bolsa de palabras resultante falla. En el emparejamiento, la verificación mediante consistencia geométrica también falla debido a que este método no tiene en cuenta la posibilidad de dichas deformaciones.

La construcción del mapa mediante un algoritmo de SLAM topológico y la localización Bayesiana son muy adecuados para una misma salida en el colon, logrando un 91 % de *recall*@8s. El principal cuello de botella del sistema es el RANSAC (Figura 4.4) para la verificación geométrica, que podría acelerarse mediante el uso de algoritmos más sofisticados como el MAGSAC++ [24, 25] adaptado a cámaras gran angular como la del endoscopio.

Los resultados recientes en aprendizaje profundo sugieren que pueden ser más robustos a deformaciones, dando lugar a sistemas más adecuados. Por ello, puede ser una buena línea de investigación futura la integración de redes neuronales en un sistema de SLAM topológico como el desarrollado, usándolas como fuente de la localización Bayesiana.

Bibliografía

[1] Laura Oliva Maza. «Navegación endoscópica con ORB-SLAM para cirugía uretral mínimamente invasiva». Trabajo Fin de Máster. EINA, Universidad de Zaragoza, 2020.

[2] Raúl Mur-Artal y Juan D. Tardós. «ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras». En: *IEEE Transactions on Robotics* 33.5 (2017), págs. 1255-1262.

[3] D.G. Lowe. «Object recognition from local scale-invariant features». En: *IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1150-1157 vol.2.

[4] Ethan Rublee y col. «ORB: An efficient alternative to SIFT or SURF». En: *IEEE International Conference on Computer Vision*. 2011, págs. 2564-2571.

[5] Pablo Fernández Alcantarilla, Jesús Nuevo y Adrien Bartoli. «Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces». En: *British Machine Vision Conference*. 2013.

[6] Daniel DeTone, Tomasz Malisiewicz y Andrew Rabinovich. «Superpoint: Self-supervised interest point detection and description». En: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, págs. 224-236.

[7] Paul-Edouard Sarlin y col. «SuperGlue: Learning Feature Matching With Graph Neural Networks». En: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, págs. 4937-4946.

[8] Alejandro Paricio García. «Reconocimiento de lugares en SLAM visual con imágenes de endoscopio». Trabajo Fin de Grado. EINA, Universidad de Zaragoza, 2021.

[9] Dorian Gálvez-López y J. D. Tardós. «Bags of Binary Words for Fast Place Recognition in Image Sequences». En: *IEEE Transactions on Robotics* 28.5 (oct. de 2012), págs. 1188-1197.

[10] Pablo Fernández Alcantarilla, Adrien Bartoli y Andrew J. Davison. «KAZE Features». En: *Computer Vision – European Conference on Computer Vision (ECCV) 2012*. Ed. por Andrew Fitzgibbon y col. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, págs. 214-227.

[11] Pablo Azagra y col. «EndoMapper dataset of complete calibrated endoscopy procedures». En: *arXiv preprint arXiv:2204.14240* (2022).

[12] Alessandro Pieropan y col. «Feature Descriptors for Tracking by Detection: a Benchmark». En: *arXiv* (2016).

[13] Rafael Muñoz-Salinas, Dorian Gálvez-López y col. *DBoW3*. 2017. URL: https://github.com/rmsalinas/DBow3 (visitado 10-08-2021).

[14] Jan Sellner. *Introduction to Fast Explicit Diffusion (FED)*. Blog. 19 de abr. de 2017. URL: https://www.jansellner.net/blog/Introduction_to_Fast_Explicit_Diffusion_(FED) (visitado 03-04-2022).

[15] P. Perona y J. Malik. «Scale-space and edge detection using anisotropic diffusion». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (1990), págs. 629-639.

[16] Joachim Weickert. «Efficient Image Segmentation Using Partial Differential Equations and Morphology». En: *Pattern Recognition* 34 (sep. de 2001), págs. 1813-1824.

[17] Martin A. Fischler y Robert C. Bolles. «Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography». En: *Commun. ACM* 24 (1981), págs. 381-395.

[18] H. C. Longuet-Higgins. «A computer algorithm for reconstructing a scene from two projections». En: *Nature* 293.5828 (ene. de 1981), págs. 133-135.

[19] Mark Cummins y Paul Newman. «Appearance-only SLAM at large scale with FAB-MAP 2.0». En: *The International Journal of Robotics Research* 30.9 (2011), págs. 1100-1123.

[20] Karel Zuiderveld. «Contrast Limited Adaptive Histogram Equalization». En: *Graphics Gems IV*. Ed. por Paul S. Heckbert. USA: Academic Press Professional, Inc., 1994, págs. 474-485.

[21] Renting Liu, Zhaorong Li y Jiaya Jia. «Image partial blur detection and classification». En: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008, págs. 1-8.

[22] Qengineering. *Blur detection with FFT in C*. 2021. URL: https://github.com/Qengineering/Blur-detection-with-FFT-in-C (visitado 05-06-2022).

[23] David G. Lowe. «Distinctive Image Features from Scale-Invariant Keypoints». En: *International Journal of Computer Vision* 60.2 (nov. de 2004), págs. 91-110.

[24] Daniel Barath, Jiri Matas y Jana Noskova. «MAGSAC: marginalizing sample consensus». En: *Conference on Computer Vision and Pattern Recognition*. 2019.

[25] Daniel Barath y col. «MAGSAC++, a fast, reliable and accurate robust estimator». En: *Conference on Computer Vision and Pattern Recognition*. 2020.

[26] Sebastian Thrun, Wolfram Burgard y Dieter Fox. *Probabilistic Robotics*. New York, NY, USA: MIT Press, 2005.

[27] Johannes Lutz Schönberger y Jan-Michael Frahm. «Structure-from-Motion Revisited». En: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[28] Johannes Lutz Schönberger y col. «Pixelwise View Selection for Unstructured Multi-View Stereo». En: *European Conference on Computer Vision (ECCV)*. 2016.

Lista de Figuras

Lista de Tablas

Anexos

Anexos A

Conjunto de datos de pruebas de AKAZE

Para las pruebas de AKAZE, se seleccionó un conjunto de pares positivos (emparejables) y pares distractores (no emparejables) automáticamente a partir de secuencias de EndoMapper. Los pares positivos son imágenes separadas entre 0,5 y 1s [11] que COLMAP [27, 28] ha identificado como dentro de un mismo clúster. Los pares distractores son imágenes que COLMAP ha identificado como pertenecientes a distintos clústers. Las figuras A.1 y A.2 muestran ejemplos de pares positivos y pares distractores del conjunto de datos.

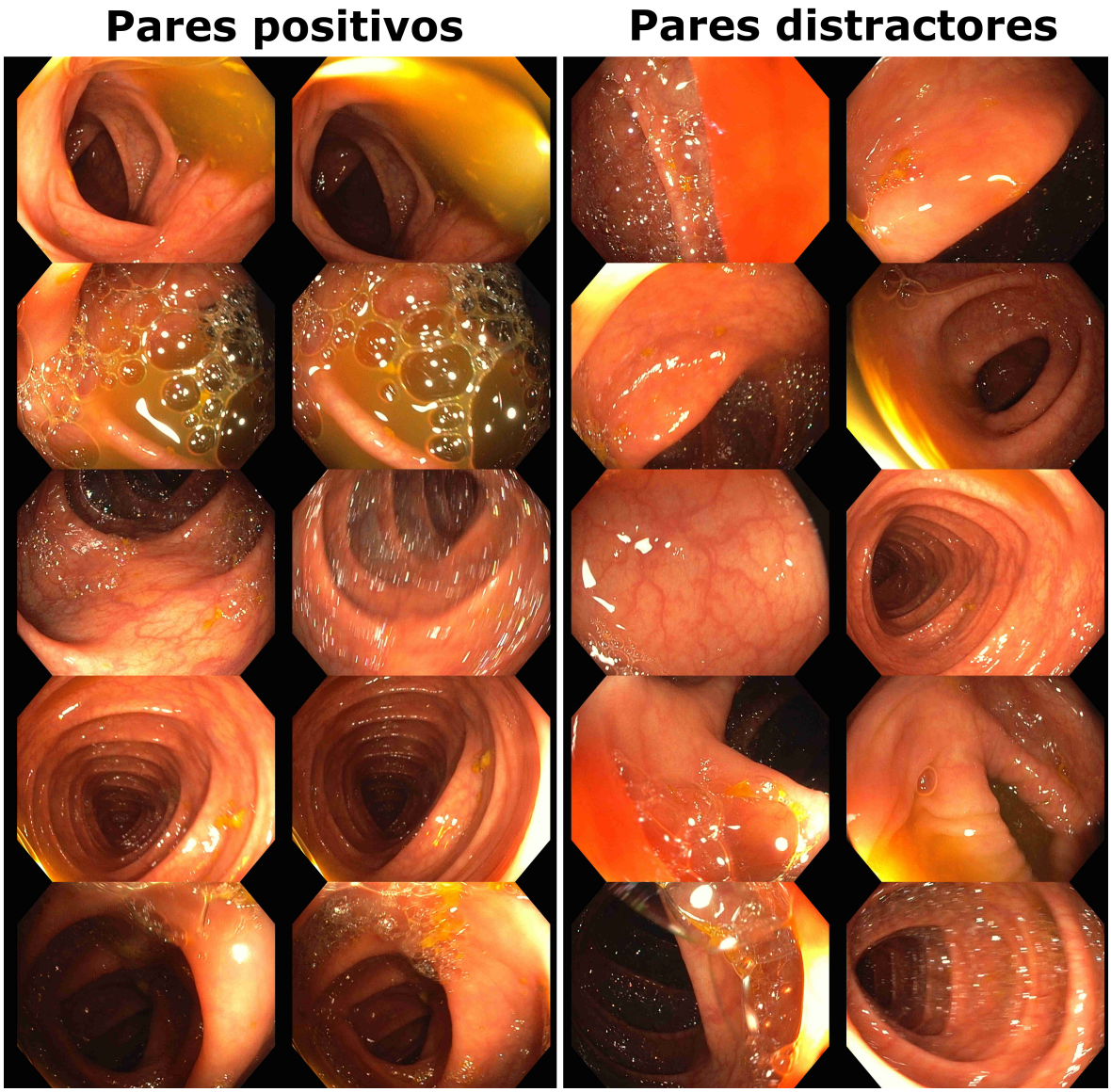


Figura A.1: Ejemplos I del conjunto de datos de pruebas de AKAZE.

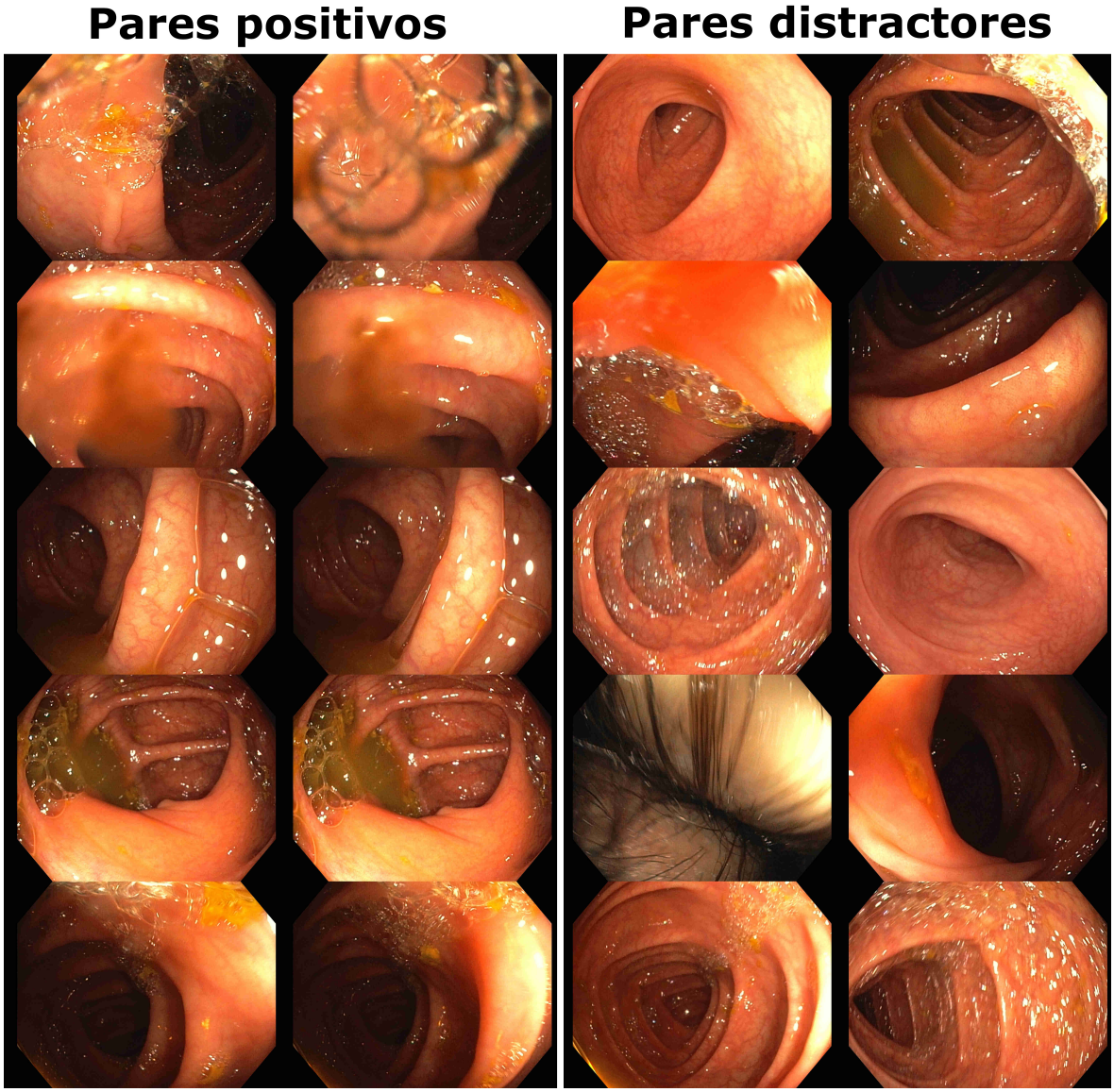


Figura A.2: Ejemplos II del conjunto de datos de pruebas de AKAZE.

Anexos B

Gestión del proyecto

El proyecto ha sido realizado durante todo el curso 2021-2022, basado en la investigación realizada mediante una Beca de Colaboración del Ministerio de Educación.

El diagrama de Gantt del proyecto se puede consultar en la figura B.1. En cada apartado de desarrollo se incluye tanto el tiempo de implementación como el de pruebas.

Las horas dedicadas a cada apartado y el total de horas invertidas en el proyecto se pueden consultar en la tabla B.1.

Tarea	**Tiempo dedicado**
Estudio	63:30
Configuración del entorno	10:15
Reuniones	34:25
Sintonización AKAZE	65:05
DBoW, resintonización y optimizaciones	169:30
SLAM topológico basado en apariencia	75:45
Redacción memoria	67:30
Total	**486:00**

Tabla B.1: Horas dedicadas a cada apartado del proyecto.

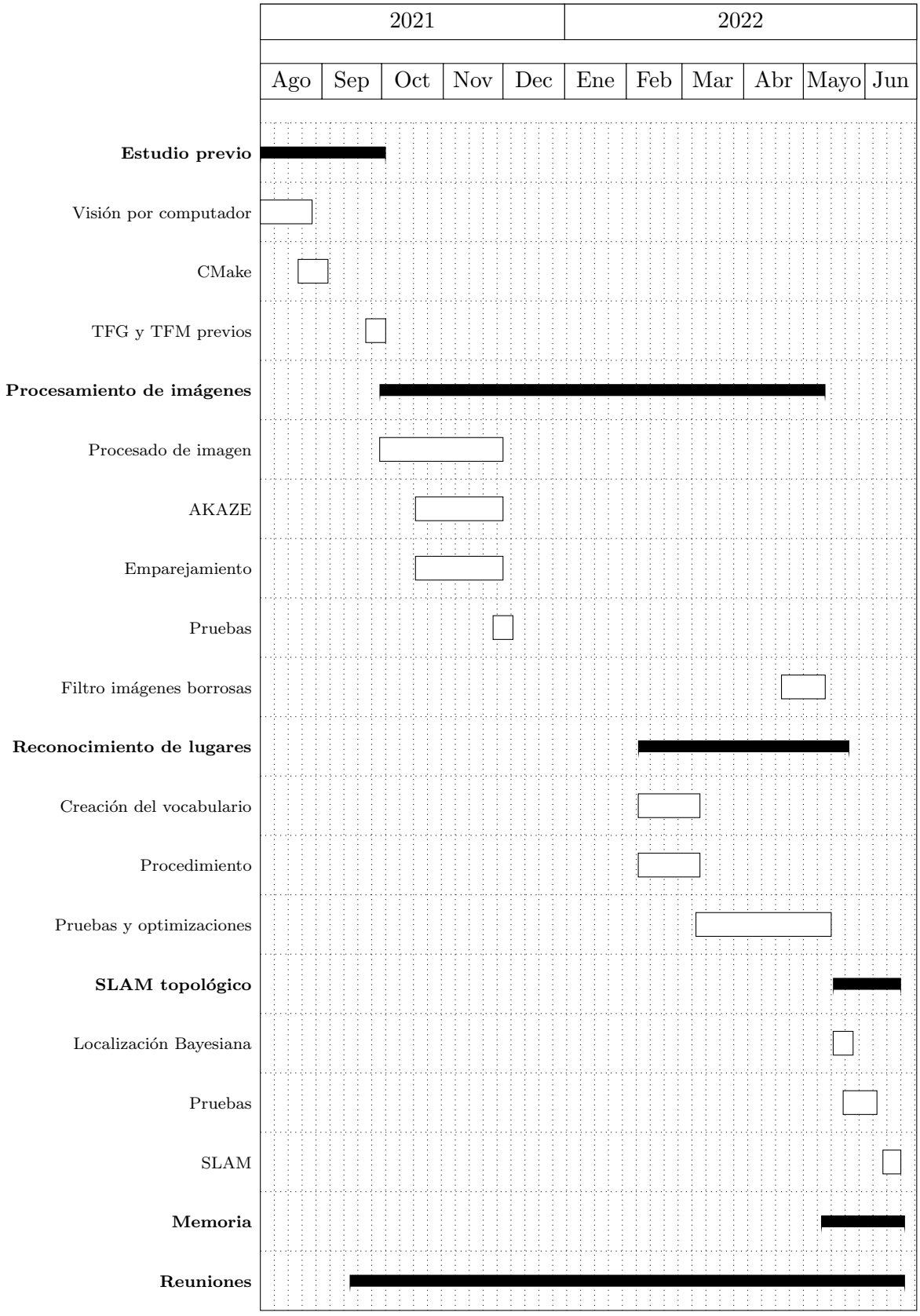


Figura B.1: Diagrama de Gantt del trabajo.