

Jornadas de Automática

Selección de keyframes para vSLAM mediante solape de barridos láser

Mengual, V., Moreno, F.-A.* , González-Jiménez, J.

Grupo de Percepción Artificial y Robótica Inteligente (MAPIR), Dept. de Ingeniería de Sistemas y Automática, Instituto Universitario en Ingeniería Mecatrónica y Sistemas Ciberfísicos (IMECH.UMA), Universidad de Málaga, Blvr. Louis Pasteur, 35, 29071 Málaga, España.

To cite this article: Mengual, V., Moreno, F.-A., González-Jiménez, J. 2024. Keyframe selection for vSLAM using laser scan overlapping. *Jornadas de Automática*, 45. <https://doi.org/10.17979/ja-cea.2024.45.10947>

Resumen

El SLAM visual se basa comúnmente en la optimización de un grafo de *keyframes*, imágenes clave en una secuencia de vídeo, para la construcción de mapas 3D y la localización de la cámara. La creación de este grafo requiere de un proceso *front-end* eficiente que seleccione *keyframes* con suficiente solape entre observaciones pero con bajo coste computacional. Este trabajo propone utilizar información de barridos láser 2D, comunes en robots móviles, para determinar el solape entre observaciones de una cámara, incorporando información sobre la estructura de la escena y seleccionando nodos distanciados entre sí cuando el solape cae bajo cierto umbral. Esto evita el costoso proceso de extracción y emparejamiento de características en las imágenes para determinar la covisibilidad entre observaciones. Las pruebas realizadas demuestran que podemos mantener el nivel de solape alrededor de un determinado umbral durante la navegación, insertando *keyframes* a diferentes frecuencias según la estructura de la escena con un coste computacional significativamente menor que con el uso exclusivo de imágenes.

Palabras clave: Robots móviles, Percepción y detección, Localización, Construcción de mapas, Navegación de robots

Keyframe selection for vSLAM using laser scan overlapping

Abstract

Visual SLAM is commonly based on optimizing a graph of *keyframes*, key images in a video sequence, for 3D map construction and camera localization. The creation of such a graph requires an efficient front-end process that selects *keyframes* with sufficient overlap between observations but with low computational cost. This work proposes to use information from 2D laser scans, common in mobile robots, to determine the overlap between observations of a camera, incorporating information about the scene structure and selecting spaced *keyframes* when the overlap falls below a certain threshold. This avoids the costly process of extracting and matching features in images to determine covisibility between observations. Our tests show that we can maintain the level of overlap around a certain threshold during navigation by inserting *keyframes* at different frequencies depending on the scene structure with significantly lower computational cost than when using images alone.

Keywords: Mobile robots, Perception and sensing, Localization, Map building, Robot Navigation

1. Introducción

Un enfoque común para el problema del SLAM visual, esto es, la construcción de mapas visuales 3D y localización de la cámara, consiste en optimizar un grafo de *keyframes*. Los *keyframes* son imágenes significativas de la secuencia de vídeo de la cámara y se representan como los nodos del gra-

fo. Los arcos entre nodos establecen la relación espacial entre estos *keyframes* y se determinan a partir del solape que ambas imágenes captan de la escena. La construcción de este grafo es una tarea que se realiza por un proceso *front-end*, con prioridad temporal, y que tiene una repercusión clave en la eficiencia y precisión de la posterior optimización del grafo.

*Autor para correspondencia: famoreno@uma.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

Esta optimización es costosa computacionalmente y se lleva a cabo en un proceso *back-end* que calcula la secuencia de poses de la cámara (su trayectoria espacial) y el mapa 3D de puntos característicos detectados en las imágenes (*keypoints*). Así, durante el proceso *front-end* de creación del grafo, hay que aplicar un procedimiento de selección de *keyframes* (nodos), que tenga un bajo coste computacional, pero que garantice un número reducido de ellos y con un grado suficiente de solape. Nótese que si el solape es débil no será posible relacionar con precisión dos *keyframes* conectados, por lo que es crucial mantener un nivel adecuado de solape. En el otro extremo, si se seleccionan dos *keyframes* representando prácticamente la misma observación de la escena, se crean nodos redundantes que aportan poco en la construcción del mapa, y se eleva la carga computacional de la optimización, que crece con el número de nodos del grafo. Nodos innecesarios, además, generan una acumulación del error de los incrementos estimados de poses, que resulta en una mayor desviación (*drift*) de la trayectoria.

En aplicaciones de interiores es habitual que un robot móvil disponga de un escáner láser de barrido radial, utilizado normalmente para detectar obstáculos y/o estimar el movimiento entre instantes de tiempo. Este trabajo presenta un método para realizar esta selección de *keyframes* aprovechando la información de este sensor para determinar el solape de las observaciones de la cámara a lo largo de la secuencia de movimiento. Esto permite escoger los nodos de la manera más distanciada posible, es decir, solo cuando el solape entre observaciones baje por debajo de un determinado umbral, incurriendo en un muy reducido coste computacional.

Es importante señalar que este trabajo se limita a secuencias de imágenes sin cierre de bucle, por lo que el objetivo del método propuesto se restringe a construir los tramos secuenciales de nodos. La revisitación de lugares para poder cerrar bucles y tener un grafo más conectado se deja como trabajo futuro.

2. Estado del arte

Los sistemas de localización visual que emplean *keyframes* se han utilizado ampliamente en robótica y visión por computador. Estos sistemas seleccionan *keyframes* basándose en métricas como la variación en la información visual y la distancia entre poses consecutivas. Métodos como ORB-SLAM en sus distintas versiones (Mur-Artal et al., 2015; Mur-Artal and Tardós, 2017) o (Zuo et al., 2017; Gomez-Ojeda et al., 2019), que incluye también líneas, son ejemplos de los que utilizan características visuales, entre otras métricas, para seleccionar y gestionar *keyframes*. A su vez, PTAM (Klein and Murray, 2007) también tiene en cuenta el cambio de pose de la cámara para la inserción de *keyframes*, mientras que (Qin et al., 2018) incorpora información inercial para realizar esta gestión. SVO (Forster et al., 2014), por el contrario, inserta un nuevo *keyframe* si la distancia entre la cámara y los restantes *keyframes* supera la profundidad media de la escena. Todos estos casos incluyen, al menos, la búsqueda de características en las imágenes para decidir sobre la inserción de nuevos *keyframes*, lo que conlleva a una cierta carga computacional.

Aunque menos habituales, existen otros trabajos como (Engel et al., 2014) que introducen nuevos *keyframes* sólo en

función del movimiento de la cámara, sin tener en cuenta la estructura de la escena. Estos métodos son muy rápidos a la hora de decidir la inserción de *keyframes* pero no se adaptan adecuadamente al entorno que están observando, pudiendo incurrir en un muestreo incorrecto de la trayectoria.

Nuestro método pretende aunar las ventajas de ambos enfoques al adaptarse a la escena y tomar la decisión de inserción con un coste computacional muy bajo.

3. Propuesta

Nuestro sistema implementa una solución rápida y eficiente a la toma de decisión sobre cuándo insertar un nuevo *keyframe* en un sistema de SLAM visual para el caso habitual de robots móviles que disponen de un escáner láser para navegación. La idea central de nuestra propuesta se basa en el cálculo del solape entre la observación del escáner láser del último *keyframe* y la del instante de tiempo actual.

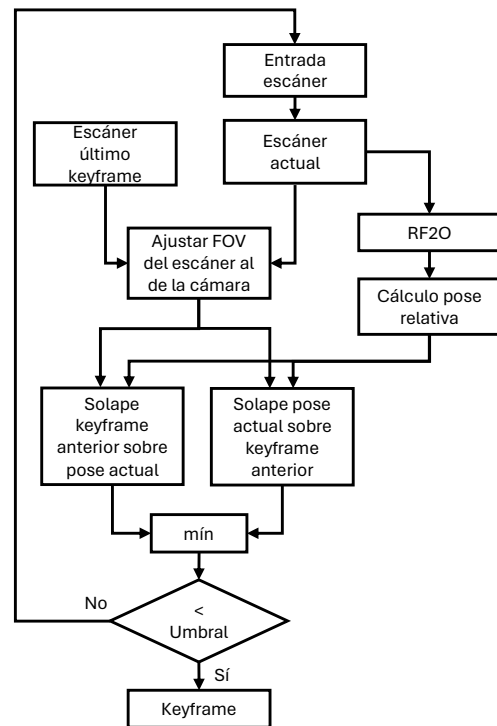


Figura 1: Pipeline del cálculo del solape y proceso de decisión para tomar *keyframes* en función del mismo.

El proceso a realizar se muestra en la Fig. 1 y comienza con la captura de una observación del escáner láser, inicializando el proceso y tomándose esta pose inicial como el primer *keyframe*. Una vez inicializado, cada vez que se reciba nueva información del láser, se procederá al cálculo del solape entre observaciones. Así, en primer lugar se realiza un ajuste del rango de ángulos permitidos del escáner láser para tener en cuenta sólo aquellas medidas que se encuentran dentro del campo de visión (*Field of View*, FoV) de la cámara. Este paso es necesario porque, aunque el FoV del escáner es sustancialmente mayor que el de la cámara, buscamos que el sistema sea adecuado para realizar SLAM visual, por lo que necesitamos que el solape medido se produzca dentro del FoV de la cámara. En el montaje considerado en este trabajo, la cámara y el

escáner láser se encuentran alineados verticalmente (ver Figura 2), por lo que el FoV del escáner láser simplemente queda restringido al FoV de la cámara.



Figura 2: Montaje del hardware. La cámara se coloca en la misma vertical que el láser, de manera que sus FoV se superpongan.

En paralelo a este proceso se determina el cambio de pose entre las observaciones utilizando la salida de RF2O (Jaimez et al., 2016), un método rápido y preciso para estimar el movimiento plano de un escáner láser a partir de escáneres de rango consecutivos. Este método realiza una alineación densa de los escáneres láser basándose en sus gradientes, obteniendo una estimación de la odometría del robot. El resultado es una aproximación de la pose absoluta del robot referida al primer instante de tiempo, lo que nos permite obtener la transformación relativa entre la pose actual y la del último *keyframe* (T en la Fig. 3).

Una vez ajustados los FoV (que denotaremos F_0 y F_1), se proyectan los puntos medidos con el láser empezando desde el ángulo mínimo del último *keyframe* (siguiendo la flecha desde el punto A en Fig. 3 medido en el sistema de referencia $\{0\}$) al sistema de referencia $\{1\}$ en el instante actual y se comprueba si el ángulo en el que se proyecta dicho punto se encuentra dentro del FoV (F_1) de la nueva pose. En caso afirmativo, se guarda el ángulo como inicio del solape y se repite el procedimiento, pero proyectando ahora los puntos empezando por el ángulo máximo (punto B en la figura). De esta forma, se obtienen los ángulos $(\hat{\theta}_{0,min}, \hat{\theta}_{0,max})$ que acotan el área de solape entre ambas poses, representado por la zona rallada en la Fig. 3:

$$\begin{aligned} \hat{\theta}_{0,min} &= \min_{\theta_0} q_{0,i} \in F_1 \\ \hat{\theta}_{0,max} &= \max_{\theta_0} q_{0,i} \in F_1, \end{aligned} \quad (1)$$

donde $q_{0,i}$, $\forall i = \{1, \dots, N\}$ es un punto arbitrario i dentro los N puntos del FoV restringido de la observación en el sistema $\{0\}$. Nótese que el valor de N se mantiene constante para todas observaciones implicadas.

A partir de estos ángulos se puede determinar el número de medidas del láser tomadas dentro de dicho área (n_0) y se calcula el solape (S_0) mediante la métrica *Intersection over Union* (IoU) ((Tanimoto, 1958)):

$$S_0 \equiv \text{IoU}_0 = \frac{n_0}{2N - n_0}. \quad (2)$$

Este proceso se repite de manera análoga pero proyectando ahora los puntos de manera inversa, es decir, los puntos medidos en la observación actual ($q_{1,i}$, $\forall i = \{1, \dots, N\}$) se proyectan sobre el sistema de referencia $\{0\}$ y se comprueba que estén contenidos en F_0 . De esta forma obtenemos otro rango de ángulos, esta vez sobre la observación actual $(\hat{\theta}_{1,min}, \hat{\theta}_{1,max})$ y otro valor de solape S_1 complementario al anterior. En este punto escogemos el menor de ambos valores, al ser más restrictivo, como el solape final entre poses. Este paso permite compensar el hecho de que el número de puntos medidos dentro de un mismo rango de ángulos es mayor cuanto más cerca esté el escáner láser de la estructura de la escena, lo que puede llevar a medidas no realistas del solape entre barridos. Por último, se comprueba si el solape final es menor a un umbral predefinido que regula la cantidad de información que queremos que las imágenes tengan en común. En caso positivo, se toma un nuevo *keyframe* en la pose actual. Se hace uso del número de medidas en lugar de la fracción de ángulo de solape puesto que pueden aparecer estructuras de la escena donde obstáculos muy cercanos llevan a resultados erróneos.

Este proceso genera un grafo donde cada nodo es un *keyframe* y los arcos conectan aquellos *keyframes* con covisibilidad, siendo su *fortaleza* el solape entre sus poses.

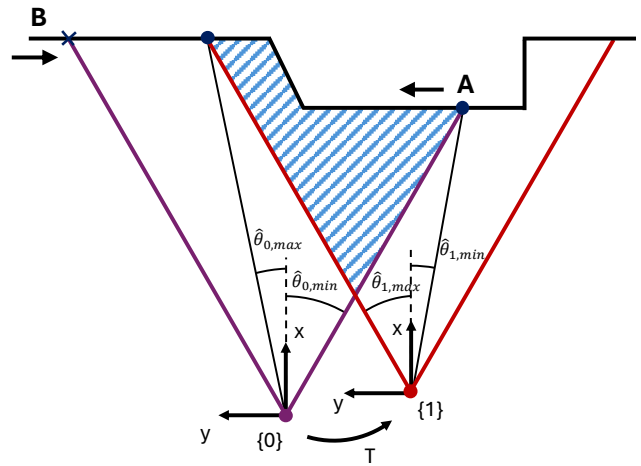


Figura 3: Solape entre poses con sistemas de referencia $\{0\}$ y $\{1\}$ y proceso de cálculo de los ángulos que acotan el área de solape. Los puntos A y B son los puntos de partida para la búsqueda de los ángulos en común. El área rallada es el solape entre ambas observaciones. La transformación T se calcula mediante RF2O.

4. Pruebas

Para evaluar la efectividad del método propuesto se han realizado dos pruebas utilizando medidas reales tomadas con un sensor Hokuyo URG-04LX-UG01, que proporciona medidas de distancia de hasta 5600 mm con un FoV de 240° y una resolución angular de 0.352° . El sensor está situado sobre la base de un robot móvil Giraff (ver Fig. 2) que también

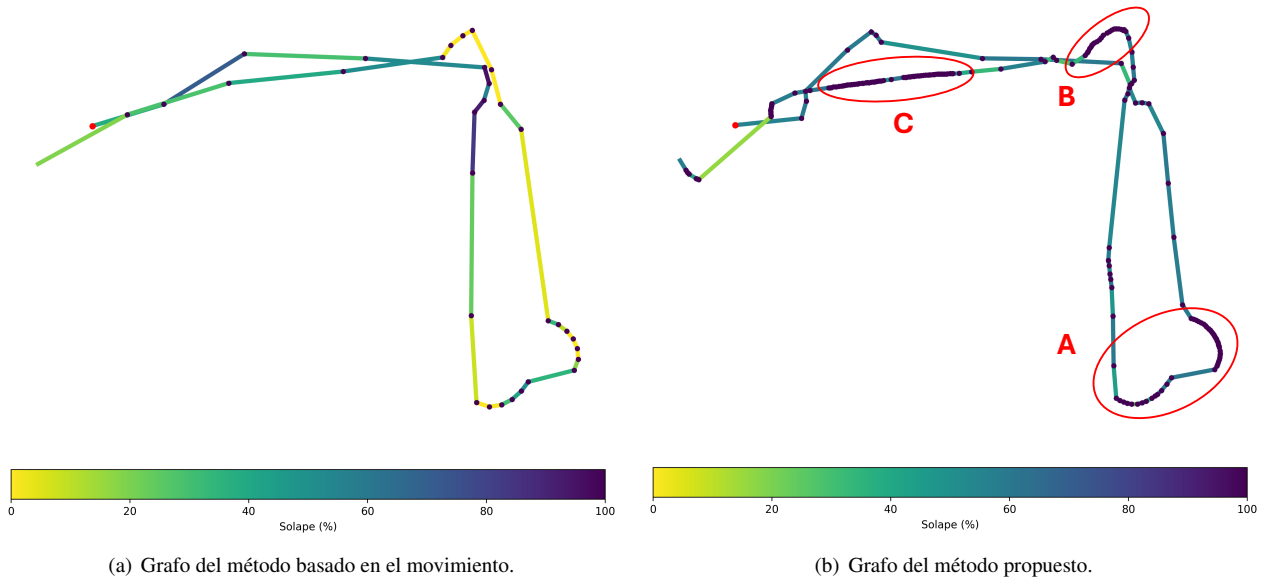


Figura 4: Grafos para el experimento del recorrido de la habitación. El punto rojo indica el inicio del recorrido, los puntos sobre la trayectoria son los *keyframes* y los colores de los arcos indican el grado de solape entre *keyframes*.

tiene instalada una cámara RGB-D Orbbec Astra que captura imágenes a 15 fps con un FoV de 60° y una resolución de 640 × 480 px. Hay que remarcar que, en este trabajo, sólo hemos usado las imágenes para ilustrar las zonas por las que pasa el robot, pero no se hace ningún procesamiento de ellas ya que todo el procedimiento se basa en las medidas del escáner láser.

En una primera prueba, el robot móvil realiza un recorrido corto y navega dentro de una habitación. La trayectoria incluye la entrada a la habitación, un giro completo alrededor de una zona de la misma, y la salida de nuevo hacia el exterior. El objetivo de esta prueba es el de testear nuestro método en el caso en el que tenemos un cambio importante en la estructura de la escena, puesto que las observaciones (escáneres láser e imágenes) cambian notablemente antes y después de pasar por el umbral de la puerta de la habitación.

Posteriormente, se ha validado nuestro método en un segundo recorrido en el que el robot navega entre dos habitaciones examinando el entorno. El propósito de este segundo experimento es determinar cómo se comporta nuestro sistema en trayectorias largas donde el robot puede estar observando una misma zona durante un largo tiempo y luego navegar a otra distinta. También buscamos identificar regiones en el grafo de poses donde, a pesar de que nuestro sistema trata de mantener un solape constante, el movimiento del robot ha provocado una bajada de solape tan abrupta que la fortaleza del arco entre dos *keyframes* consecutivos es muy baja. Estos lugares en el grafo son potenciales candidatos para realizar un corte de grafo y comenzar otro mapa.

En ambos experimentos hemos comparado nuestro selector de *keyframes* con un algoritmo basado únicamente en la distancia recorrida o la rotación realizada por el robot entre el último *keyframe* y el instante actual. Para ambos métodos, se ha empleado la métrica propuesta en la Eq. (2) para el cálculo del solape.

En el primer experimento obtenemos los grafos de *key-*

frames mostrados en la Fig. 4, donde se pueden apreciar las diferencias existentes entre ambos métodos para el recorrido corto. Los colores de los arcos entre *keyframes* representan la fortaleza de los mismos (es decir, el solape entre ellos), siendo amarillos aquellos más débiles y azules los que tienen mayor solape. Como se puede apreciar, nuestro método consigue mantener un solape entre *keyframes* elevado y prácticamente constante, evitando así la redundancia en la toma de *keyframes* y mejorando la estructura del grafo. Además, puede verse cómo se muestrea a mayor frecuencia en aquellas zonas donde el robot hace movimientos más rápidos y las regiones que presentan una estructura más compleja. Esto es consistente con el comportamiento esperado, ya que en estas situaciones el entorno cambia rápidamente y es necesario capturar más información para una representación precisa. Por el contrario, para el método basado en movimiento, al no tener en cuenta la información del entorno, el solape entre *keyframes* varía mucho dependiendo de si el robot está cambiando de habitación, es decir, si entra o sale, o si se realiza un giro con obstáculos muy cercanos.

En la Fig. 4(b) se pueden apreciar tres zonas de interés, marcadas con A, B y C. Estas mismas zonas pueden verse también en la Fig. 5, donde se representa el solape calculado a lo largo del tiempo por ambos métodos (en azul el método basado en movimiento y en naranja nuestra propuesta). Los puntos verdes y rojos, respectivamente, representan aquellos *frames* que se han considerado *keyframes* en ambos métodos.

La región A representa una zona con varias rotaciones del robot dentro de la habitación. En las figuras se puede apreciar cómo ambos métodos introducen un mayor número de *keyframes* en el grafo ya que, por un lado, la rotación es tenida en cuenta por el primer método y el solape desciende rápidamente debido al movimiento. En cualquier caso, nuestro sistema introduce un número mayor de *keyframes* para mantener constante el solape (en este experimento el umbral escogido es

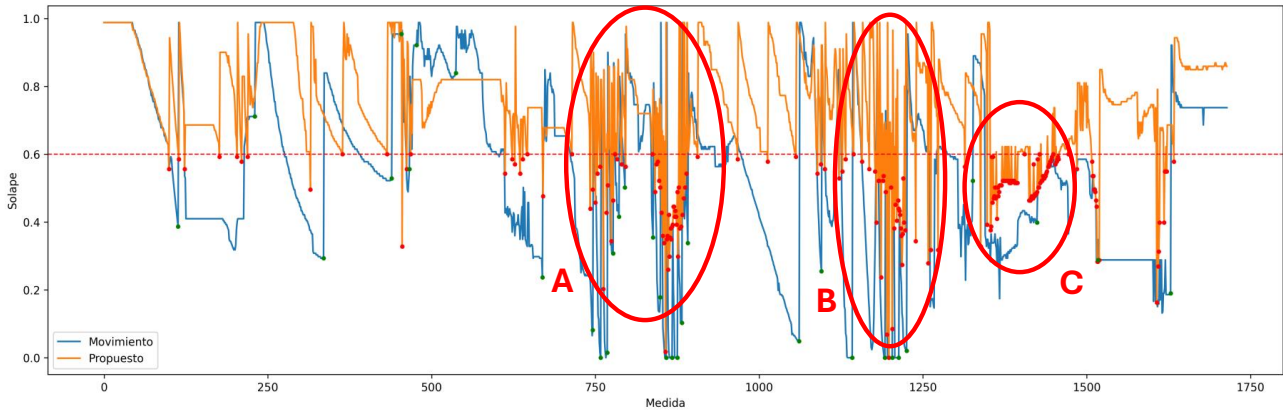


Figura 5: Comparación del solape entre ambos métodos para el primer experimento: basado en movimiento (azul) y nuestra propuesta (naranja). Los *keyframes* insertados se marcan con puntos verdes y rojos, respectivamente. Las zonas A, B y C son tramos representativos en cuanto a las diferencias entre ambos métodos.

del 60 %). De manera similar, la zona B es otra región de giro pronunciado en la que el robot está orientándose para salir de la habitación. La región C, por el contrario, presenta un caso interesante ya que se trata del robot atravesando el umbral de una puerta. Es en esta situación donde se muestran mayores diferencias entre ambos métodos, pudiéndose apreciar cómo nuestra propuesta muestrea considerablemente esta zona al haber un cambio importante en la escena, mientras que el método basado en movimiento no, reduciendo, de esta forma, el solape entre observaciones. La Fig. 6 muestra dos imágenes de ejemplo de esta zona.



Figura 6: Ejemplo de imágenes en la zona C.

Los resultados del segundo experimento (ver Fig. 7 y Fig. 8) muestran, de nuevo, que nuestro método mantiene un solape casi constante a lo largo de una trayectoria mayor al experimento anterior. Sólo encontramos una zona con bajo solape en torno a la observación 1200 donde se aprecia, en la Fig. 8, que para ambos métodos cae hasta el 20 %, fruto de un movimiento muy brusco. Esta situación se refleja en el grafo con el arco de color amarillo que presenta nuestro método.

Estas dos pruebas validan nuestra propuesta en comparación con un método que no tiene en cuenta la estructura de la escena, demostrando un mejor rendimiento en zonas con cambios importantes de escena como transiciones entre habitaciones. Sin embargo, muchos de los métodos existentes sí introducen métricas de similitud entre *keyframes* para decidir su inserción en el sistema, aunque la mayoría de ellos están basados en la búsqueda de puntos de interés en las imágenes y su posterior emparejamiento para medir la covisibilidad. Este enfoque, aunque robusto, puede llegar a ser costoso computacionalmente. Nuestro método, en cambio, es muy ligero ya que sólo necesita proyectar unos pocos puntos en dos sistemas de

referencia y determinar el número de puntos solapados entre dos observaciones de escáner láser. Para conocer la diferencia de velocidad de cálculo entre el método propuesto y los habituales empleados por los *front-ends* de los métodos de SLAM visual, se ha calculado la media del tiempo de procesamiento de los datos, su varianza y su mediana para nuestra propuesta (el cálculo del solape) y la extracción y emparejamiento de los puntos de interés con ORB y SIFT (ver Tabla 1). Los resultados demuestran la poca carga computacional que supone el cálculo del solape con nuestro método en comparación.

Tabla 1: Mediana, media y varianza del tiempo de procesamiento de los datos.

	$\mu_{1/2}$ (ms)	μ (ms)	σ^2 (ms)
ORB	20.78	27.95	0.16
SIFT	158.16	161.08	0.61
Propuesto	1.72	2.56	1.22

5. Conclusiones y trabajos futuros

En este trabajo hemos presentado un método para seleccionar *keyframes* de forma eficiente que utiliza barridos láser 2D para estimar el solape entre observaciones de una cámara sin necesidad de procesar las imágenes. Nuestra propuesta aprovecha la habitual presencia de escáneres láser radiales en robots móviles para, con un coste computacional muy reducido, incorporar la información de la estructura del entorno a la selección de *keyframes* para la construcción de grafos en sistemas de SLAM visual.

De esta forma, hacemos uso de la estimación de odometría del método RF2O para determinar la pose relativa entre el último *keyframe* y la pose actual y la utiliza para proyectar los puntos observados en un barrido en el sistema de referencia del otro, obteniéndose de esta manera el rango de ángulos comunes entre ambas observaciones. Utilizando la métrica *Intersection over Union* con el número de puntos en común entre observaciones obtenemos un porcentaje de solape entre ambas. Cuando este solape cae por debajo de un determinado umbral se introduce un nuevo *keyframe* en el sistema.

Las pruebas realizadas demuestran que el método propuesto es efectivo para mantener un solape aproximadamente uniforme entre los *keyframes*. Esto asegura una construcción

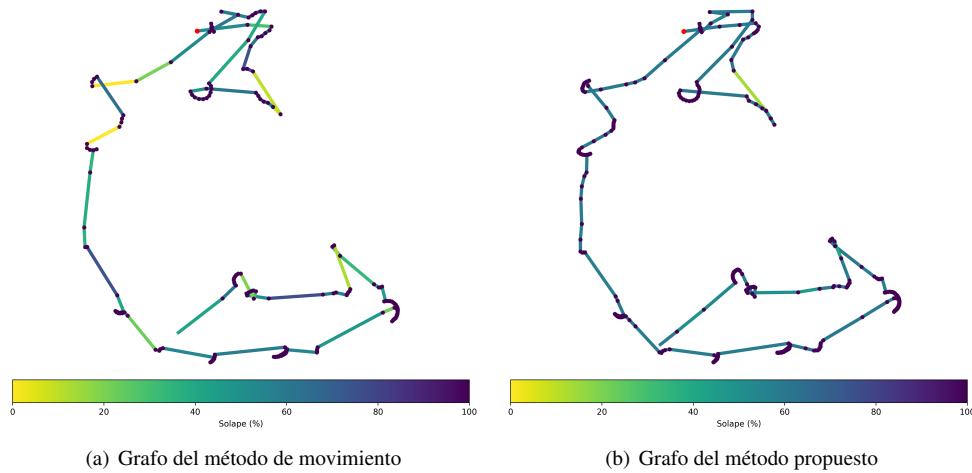


Figura 7: Grafos para el experimento del recorrido de varias habitaciones. El punto rojo indica el inicio del recorrido, los puntos sobre la trayectoria son los *keyframes* y los colores de los arcos indican el grado de solape entre *keyframes*.

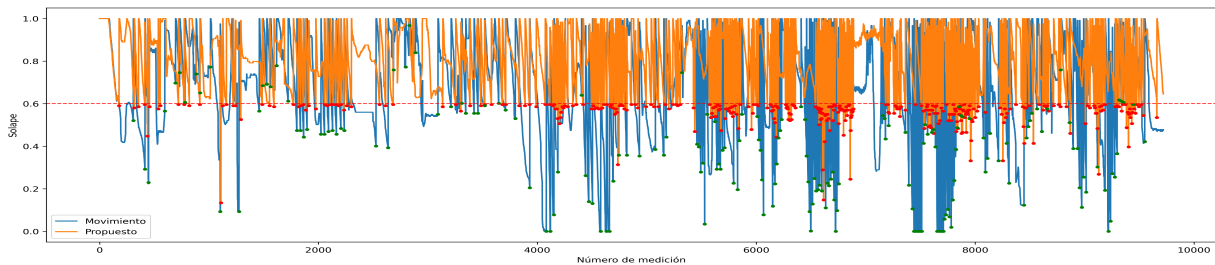


Figura 8: Comparación del solape entre ambos métodos para el segundo experimento: basado en movimiento (azul) y nuestra propuesta (naranja). Los *keyframes* insertados se marcan con puntos verdes y rojos, respectivamente.

eficiente del grafo de *keyframes* para SLAM visual, reduciendo la redundancia e incorporándolos a mayor frecuencia en zonas donde la estructura es más cambiante como en giros bruscos, cambio de ambientes (p.ej. paso de una habitación a otra) y ante la presencia de obstáculos cercanos. Además, el coste computacional es sensiblemente menor al incurrido cuando se utiliza la información de las imágenes para introducir *keyframes* en el sistema.

Partiendo de este trabajo preliminar, en el futuro se prevé abordar la integración del método en un sistema completo de SLAM visual, como ayuda al *front-end*, que permita comparar la calidad del mapa final con los producidos por otros métodos. Además, se prevé la identificación de regiones en el grafo para realizar cortes de grafo en zonas con poco solape.

Agradecimientos

Este trabajo ha sido desarrollado en el contexto de los proyectos ARPEGGIO (PID2020-117057GB-I00) y Voxeland (JA.B1-09), financiados por el Ministerio de Ciencia e Innovación y la Universidad de Málaga, respectivamente.

Referencias

Engel, J., Schöps, T., Cremers, D., 2014. LSD-SLAM: Large-scale direct monocular SLAM. In: European conference on computer vision. Springer, pp. 834–849.
 DOI: 10.1007/978-3-319-10605-2_54

Forster, C., Pizzoli, M., Scaramuzza, D., 2014. Svo: Fast semi-direct monocular visual odometry. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE, pp. 15–22.
 DOI: 10.1109/ICRA.2014.6906584

Gomez-Ojeda, R., Moreno, F.-A., Zuniga-Noël, D., Scaramuzza, D., Gonzalez-Jimenez, J., 2019. PL-SLAM: A stereo SLAM system through the combination of points and line segments. IEEE Transactions on Robotics 35 (3), 734–746.
 DOI: 10.1109/TR0.2019.2899783

Jaimez, M., Monroy, J. G., Gonzalez-Jimenez, J., 2016. Planar odometry from a radial laser scanner. a range flow-based approach. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 4479–4485.
 DOI: 10.1109/ICRA.2016.7487647

Klein, G., Murray, D., 2007. Parallel tracking and mapping for small AR workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE, pp. 225–234.
 DOI: 10.1109/ISMAR.2007.4538852

Mur-Artal, R., Montiel, J. M. M., Tardos, J. D., 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE transactions on robotics 31 (5), 1147–1163.
 DOI: 10.1109/TR0.2015.2463671

Mur-Artal, R., Tardós, J. D., 2017. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Transactions on Robotics 33 (5), 1255–1262.
 DOI: 10.1109/TR0.2017.2705103

Qin, T., Li, P., Shen, S., 2018. Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics 34 (4), 1004–1020.
 DOI: 10.1109/TR0.2018.2853729

Tanimoto, T. T., 1958. Elementary mathematical theory of classification and prediction.

Zuo, X., Xie, X., Liu, Y., Huang, G., 2017. Robust visual SLAM with point and line features. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 1775–1782.
 DOI: 10.1109/IROS.2017.8205991