

Universidad Miguel Hernández de Elche
MÁSTER UNIVERSITARIO EN ROBÓTICA



”Desarrollo de métodos de localización probabilística jerárquica basado en información visual omnidireccional”

Trabajo de Fin de Máster

Curso académico 2022-2023

Autor: Raúl Bezares Pino

Tutores: Luis Payá Castelló
Sergio Cebollada López

Índice general

Índice de figuras	IV
Índice de tablas	VII
1 Introducción	1
1.1 Localización en robótica móvil	3
1.1.1 Sensores	3
1.1.2 Descriptores holísticos	6
1.2 Inteligencia Artificial	7
1.3 Motivación	8
1.4 Estructura de la memoria	9
2 Estado del arte	10
2.1 Localización jerárquica	10
2.2 Localización probabilística	11
2.3 Inteligencia artificial	13
3 Metodologías	15
3.1 Localización batch	17
3.2 Localización jerárquica	18
3.3 Localización probabilística con información métrica de baja confianza	19
3.4 Localización jerárquica probabilística	21
3.4.1 Entrenamiento CNN	21
3.5 Localización probabilística con información métrica de confianza	22
3.6 Localización probabilística sin información métrica	24
4 Experimentos y resultados	26
4.1 Experimentos previos	26
4.2 Experimentos de localización	28
4.2.1 Localización batch	28
4.2.2 Localización jerárquica	29
4.2.3 Localización probabilística basada en filtro bayesiano	30
4.2.4 Localización jerárquica probabilística	34
4.2.5 Localización probabilística con y sin métrica	34

5 Conclusiones y trabajos futuros	37
5.1 Conclusiones	37
5.2 Trabajos futuros	38
Bibliografía	40



Índice de figuras

1-1	Diferentes configuraciones de robots móviles.	1
1-2	Cámara e imagen omnidireccional.	5
3-1	Proceso de mapeado.	15
3-2	Mapa Friburgo. Fuente [45]	16
3-3	Diagrama de flujo del algoritmo de búsqueda del vecino más cercano.	17
3-4	Diagrama de flujo del algoritmo jerárquico.	18
3-5	Diagrama de flujo del algoritmo probabilístico con información métrica de baja confianza.	19
3-6	Diagrama de flujo del algoritmo jerárquico probabilístico.	20
3-7	Ejemplos del umbral dual de búsqueda.	22
3-8	Diagrama de flujo del algoritmo con información métrica de confianza.	23
3-9	Ventana de búsqueda del método sin información métrica.	24
3-10	Comparativa en caminos cruzados.	25
3-11	Diagrama de flujo del algoritmo sin información métrica.	25
4-1	Matrices de confusión.	26
4-2	Resultados del método del vecino más cercano.	28
4-3	Histogramas de proximidad de imagen seleccionada.	29
4-4	Resultados del método jerárquico.	30
4-5	Mapa de Friburgo separado en 50 clusters.	31
4-6	Mapa rejilla. Fuente [51]	32
4-7	Resultados del método probabilístico. Mapa de trayectoria.	33
4-8	Resultados del método probabilístico. Mapa de rejilla.	33
4-9	Resultados del método jerárquico probabilístico.	34
4-10	Comparativa de resultados de los métodos con y sin información métrica.	35
4-11	Comparativa de resultados de los métodos con y sin información métrica. Detalle.	36

Índice de tablas

4-1 Métricas de entrenamiento CNN 27



1 Introducción

La robótica es un campo multidisciplinario en constante evolución que combina la ingeniería, la inteligencia artificial y la ciencia de la computación. Su objetivo es diseñar, desarrollar y operar sistemas autónomos capaces de interactuar inteligentemente con el entorno y llevar a cabo tareas específicas. Los robots, como ejemplos de estos sistemas, han adquirido una relevancia significativa en una amplia gama de aplicaciones, desde la automatización industrial [1] hasta la exploración espacial[2].

Dentro del ámbito de la robótica, la robótica móvil ocupa un lugar destacado. Se refiere al estudio y desarrollo de robots capaces de moverse y operar en entornos variables y dinámicos. En contraste con los robots estáticos, que están diseñados principalmente para tareas repetibles y precisas, los robots móviles tienen la capacidad de desplazarse para llevar a cabo la tarea para la cual han sido diseñados, lo que les permite interactuar con su entorno de manera más dinámica y versátil. Estos robots encuentran aplicaciones en áreas como la logística [3], la agricultura [4] o la exploración submarina [5], entre otras. En la actualidad, se ha incrementado notablemente el uso de robots móviles autónomos en estos campos gracias a la mejora de eficiencia que proporcionan en automatización de procesos sin suponer un peligro para las personas que se encuentren en su entorno. Por ello, se disponen de muchas configuraciones diferentes, como se ve en la figura 1-1

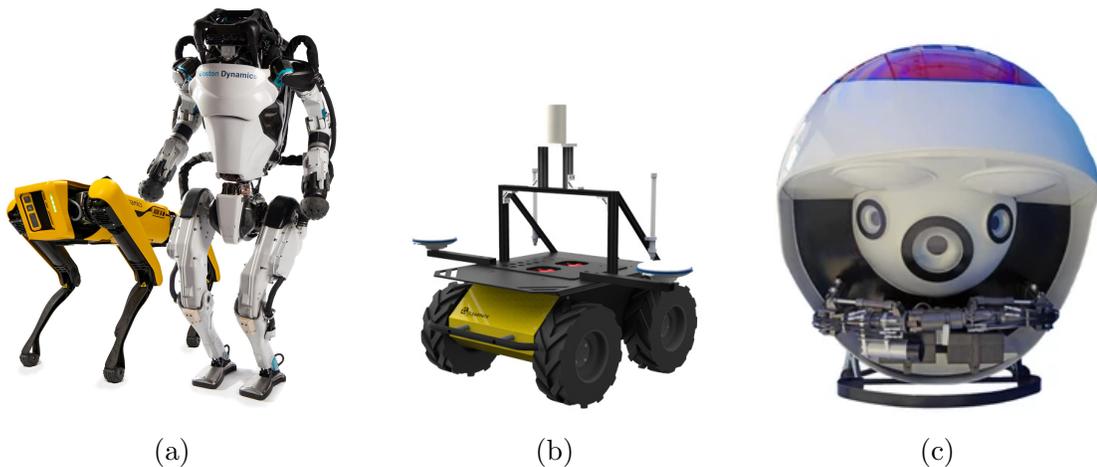


Figura 1-1: Diferentes configuraciones de robots móviles.

La robótica móvil sigue en constante evolución y desarrollo desde los primeros trabajos propuestos [6]. Con el avance de los algoritmos y la capacidad de los sistemas de percepción

y computación, la integración de la función de visión en los robots móviles se ha convertido en una de las herramientas más importantes para realizar tareas de mapeo, localización y navegación. La habilidad de percepción la proporcionan los sensores, entre los cuales destacan los de visión por la gran cantidad de información que pueden capturar, útil para realizar tareas muy diversas, lo que ha llevado a la utilización de la percepción visual como una técnica fundamental en la realización de tareas robóticas. La robótica visual se basa en métodos de visión por computador que permiten al robot comprender la escena y los objetos que se encuentran en ella de forma parecida a los humanos; dotándole por tanto, de la capacidad de tomar decisiones y realizar acciones en consecuencia.

La robótica móvil presenta desafíos únicos debido a esta necesidad de adquirir información del entorno, tomar decisiones en tiempo real y desplazarse de manera autónoma. Para abordar estas tareas, los robots móviles dependen de los sistemas sensoriales, de algoritmos de control y técnicas de localización, mapeo y navegación. El desarrollo de estos componentes es fundamental para lograr un funcionamiento eficiente y seguro de los robots móviles en toda clase de entornos, que suelen ser no estructurados y con condiciones impredecibles. Estos entornos comprenden desde espacios interiores con obstáculos y cambios imprevistos, hasta entornos exteriores con terrenos complejos y factores ambientales variables. En muchos casos, los robots móviles son desplegados en entornos peligrosos o inaccesibles para los seres humanos, como en operaciones de búsqueda y rescate [7], exploración espacial o inspección de infraestructuras [8].

Como se ha mencionado antes, para lograr la navegación autónoma de los robots móviles, es necesario resolver una serie de tareas. Es de especial interés la tarea de localización en robótica móvil, debido a su impacto en las otras tareas clave, el mapeo y el SLAM (siglas en inglés de *Simultaneous Localization and Mapping*). El mapeo se refiere a la creación de un modelo del entorno circundante, capturando la geometría, las características y la estructura del mismo. Una vez se ha obtenido una representación del entorno mediante el proceso de mapeo, se puede llevar a cabo la tarea de localización, que consiste en que el robot conozca su propia posición en relación con el entorno.

El SLAM es un enfoque más desafiante, ya que implica la construcción de un mapa del entorno mientras el robot se localiza simultáneamente en él. Es una tarea compleja que requiere una fusión de datos de sensores, algoritmos de estimación de posición y mapeo, y técnicas de optimización. El SLAM es especialmente relevante en escenarios en los que el entorno no se conoce de antemano, como en la exploración de áreas desconocidas o en la navegación en entornos dinámicos donde los objetos pueden moverse.

En resumen, la tarea de localización en robótica móvil es un pilar fundamental para el funcionamiento autónomo y efectivo de los robots. Con una localización confiable y precisa, los robots móviles pueden realizar tareas de manera segura y eficiente, adaptándose a cambios en el entorno y navegando de manera inteligente.

Para lograr este objetivo, el presente trabajo propone la utilización de técnicas basadas en descriptores holísticos, que capturan características globales de la escena, para lo cual se

hará uso de cámaras omnidireccionales, que permiten obtener información de todo el entorno que rodea al robot. También se explorará el uso de técnicas de inteligencia artificial para mejorar la capacidad de adaptación y toma de decisiones del robot, tanto para la obtención de descriptores holísticos como en otras etapas de la resolución del problema de localización. Se realizarán experimentos utilizando datos reales para evaluar el desempeño de los algoritmos propuestos y se compararán con enfoques existentes en la literatura.

1.1. Localización en robótica móvil

En el contexto de la robótica móvil, la localización se refiere a la estimación de la posición y orientación del robot en relación con un sistema de coordenadas de referencia, el mapa o modelo del entorno. La localización es una tarea fundamental en el campo de la robótica móvil, que desempeña un papel crítico en el funcionamiento autónomo y eficiente de los robots en entornos dinámicos y variables.

Para lograr esta flexibilidad, es fundamental que el robot disponga de un modelo del entorno en el que se encuentra. Un modelo del entorno es una representación interna del espacio circundante, que incluye información sobre la geometría, los objetos presentes, las características del terreno y otros aspectos relevantes para la navegación del robot. La localización entonces se refiere a la capacidad del robot para estimar su posición y orientación dentro de este entorno, que se supone conocido. Así, es posible lograr una localización precisa comparando la información percibida por los sensores con el modelo, de modo que el robot sea capaz de tomar decisiones informadas y planificar rutas óptimas para alcanzar sus objetivos.

Estos métodos pueden dar como resultado la posición que el robot ocupa dentro del modelo del entorno, o una distribución de probabilidades de que el robot ocupe cada posición del entorno, según cómo se haya construido el modelo y diseñado el algoritmo de localización. También es posible hacer una localización jerárquica, en la que primero se estima la zona general donde es más probable que se encuentre el robot y a continuación se afina la posición final.

1.1.1. Sensores

Para llevar a cabo la tarea de localización, los sensores desempeñan un papel crucial al proporcionar información clave para la tarea de localización precisa. Existe una variedad de estos dispositivos, que se pueden clasificar en exteroceptivos, si capturan datos del entorno circundante, y propioceptivos, que obtienen información del propio robot. La combinación de ambos permite al robot adquirir conocimiento y comprensión de su posición, orientación y las características del entorno en el que se encuentra.

El sistema de posicionamiento global (GPS, por sus siglas en inglés) es un sistema de navegación por satélite ampliamente utilizado en aplicaciones de localización. Utiliza una constelación de satélites en órbita para determinar la posición geográfica precisa del receptor

GPS. El GPS proporciona coordenadas de latitud, longitud y altitud, lo que permite una localización global del robot en el planeta, que puede servir como punto de partida para muchos métodos de localización en robótica móvil [9]. Sin embargo, el GPS tiene limitaciones en entornos interiores o en áreas con obstáculos que obstruyen la señal de los satélites, por lo que su uso queda restringido a determinadas aplicaciones en exteriores.

El LiDAR (Light Detection and Ranging) es un sensor óptico que emite pulsos de luz láser y mide el tiempo que tarda en recibir los reflejos de esos pulsos. Esto permite construir un mapa tridimensional de la superficie y estructuras circundantes al robot. El LiDAR proporciona información muy detallada sobre la geometría del entorno y la ubicación de los objetos. Gracias a esto, es especialmente útil para tareas de localización [10], como la detección de obstáculos y la creación de mapas precisos del entorno, especialmente en entornos de interior. Sin embargo, si el entorno es altamente estructurado y predecible, es posible que la información proporcionada por un LiDAR detallado no sea fundamental para la tarea de localización, además de que estos dispositivos pueden tener costos prohibitivos; de modo que es necesario considerar bien todos los factores antes de seleccionar este dispositivo.

Los encoders son sensores utilizados para medir la velocidad y el desplazamiento de las ruedas o articulaciones del robot. Registran el número de rotaciones o pulsos generados por las ruedas o los motores, lo que permite estimar la distancia recorrida y el cambio en la orientación del robot. Los encoders son esenciales para la odometría, que consiste en estimar la posición del robot a partir de sus movimientos y se utiliza como fuente de información para los algoritmos de localización [11], aunque a menudo no ofrecen información demasiado certera por sí mismo, ya que no puede tener en cuenta cómo el robot realmente interactúa con el entorno, por lo que se suele usar como fuente de información adicional en algoritmos de fusión de sensores.

Los sensores más ampliamente utilizados son los sensores visuales, las cámaras, que capturan imágenes del entorno circundante. Proporcionan información visual rica sobre la escena, incluyendo la geometría, los colores, las texturas y la posición relativa de los objetos. En la localización, las cámaras se emplean para realizar tareas como el reconocimiento de características, el emparejamiento de imágenes y la estimación de la posición y orientación del robot mediante técnicas de visión por computadora. Dado que se puede captar una gran variedad de información, que puede ser usada de distintas formas en cada algoritmo, según qué características sean más distintivas en el entorno de trabajo [12]. Del mismo modo, estos sensores también son muy sensibles a esta variabilidad de la información que puede ser captada, como los cambios de iluminación en la escena, de modo que es imperante desarrollar algoritmos robustos a estos cambios.

De entre los distintos tipos de cámaras con diferentes características, son de especial interés las **cámaras omnidireccionales**, que se presentan como una opción prometedora debido a su capacidad para capturar un campo de visión completo alrededor del robot. A diferencia de las cámaras convencionales con campos de visión más reducidos, las cámaras omnidireccionales ofrecen una visión global y envolvente del entorno, lo que puede resultar

ventajoso en diversas aplicaciones de localización, tal y como se puede observar en la figura 1-2.

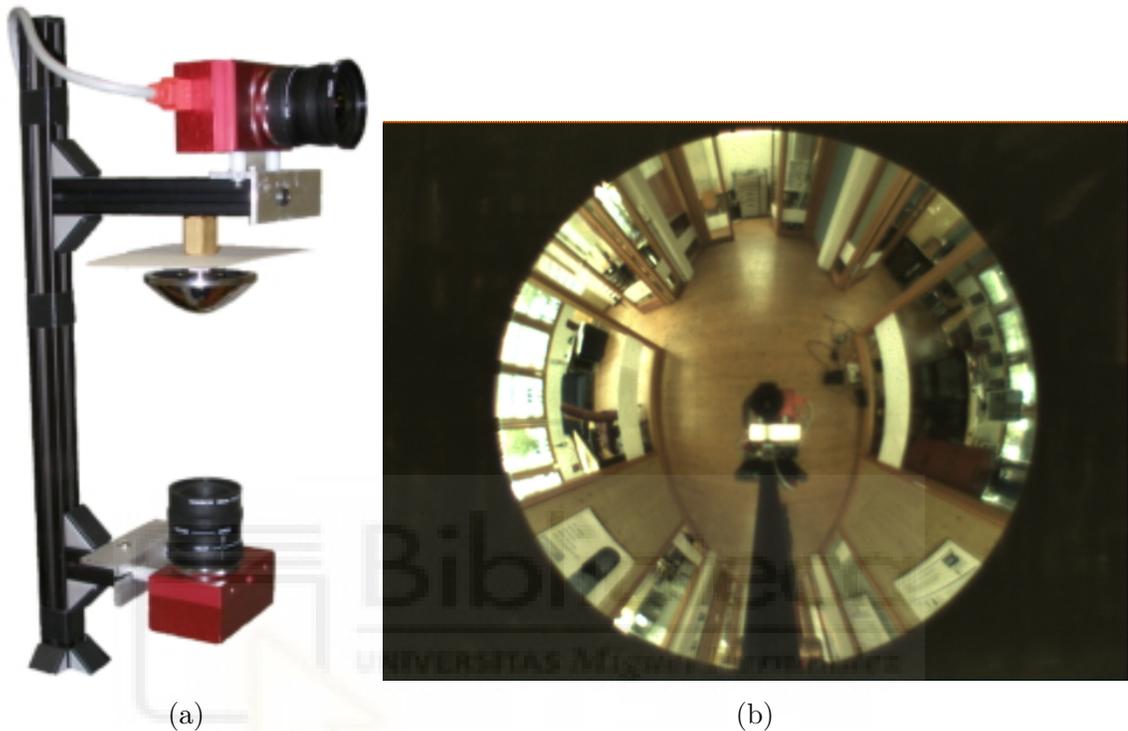


Figura 1-2: Cámara e imagen omnidireccional.

Una de las principales ventajas de las cámaras omnidireccionales es su capacidad para reducir los problemas de oclusión o cambios de iluminación [13]. Al abarcar un campo de visión más amplio, estas cámaras permiten capturar información de áreas que de otro modo podrían estar bloqueadas o fuera del alcance de las cámaras convencionales, superando estas restricciones mencionadas. Esto es especialmente útil en entornos variables con obstáculos, espacios reducidos o estructuras arquitectónicas complejas.

Además, las cámaras omnidireccionales facilitan la detección y seguimiento de objetos en movimiento, ya que no requieren una orientación precisa del robot para capturar información visual. Esto puede ser especialmente útil en aplicaciones de seguimiento de objetos en tiempo real [14], como el rastreo de personas [15], vehículos u otros robots.

Otro aspecto a destacar es la capacidad de las cámaras omnidireccionales para proporcionar información de contexto adicional. Al capturar un campo de visión completo, estas cámaras permiten analizar la distribución espacial de objetos, características del entorno y relaciones espaciales entre ellos. Esta información contextual puede ser aprovechada para mejorar la precisión y la robustez de los algoritmos de localización, así como para la tarea complementaria del mapeo [16].

En términos de procesamiento de imágenes, el uso de cámaras omnidireccionales implica

retos adicionales debido a la distorsión inherente a su sistema óptico. Sin embargo, existen técnicas de calibración y corrección de imagen específicas para cámaras omnidireccionales que permiten obtener imágenes rectificadas y facilitan el procesamiento posterior.

La elección y combinación de los sensores depende del entorno de trabajo y de los requisitos específicos de la tarea de localización. Al integrar múltiples sensores, se pueden obtener estimaciones más precisas y robustas de la posición del robot. Los avances en la tecnología de sensores y la fusión de datos sensoriales continúan impulsando la mejora de la localización en robótica móvil, permitiendo aplicaciones más sofisticadas y autónomas.

1.1.2. Descriptores holísticos

Un descriptor holístico, o descriptor de apariencia global, es una representación compacta de una imagen que trata de representar las características esenciales de la escena en su conjunto, por tanto, esta técnica puede resultar de interés para llevar a cabo la tarea de localización, ya que al capturar características globales de la escena, son especialmente útiles para lograr una localización precisa y robusta, además de permitir una comparación directa entre imágenes.

La obtención de descriptores holísticos implica el uso de técnicas de procesamiento de imágenes y, en algunos casos, el uso de aprendizaje automático. En primer lugar, se recopila un conjunto de imágenes representativas del entorno en el que el robot operará. Estas imágenes pueden ser capturadas por cámaras, lidar u otros sensores utilizados en el sistema de percepción del robot, de modo que se puede obtener una variedad de descriptores diferentes, adecuados según las características a analizar.

El descriptor HOG (*Histogram of Oriented Gradients*) se basa en la distribución espacial de los gradientes de intensidad en una imagen. El proceso implica dividir la imagen en celdas pequeñas y calcular el histograma de las orientaciones de los gradientes en cada celda. Luego, los histogramas se concatenan para formar un vector de características que describe la estructura y textura de la imagen. El descriptor HOG es robusto a cambios de iluminación y proporciona información sobre la forma y los bordes de los objetos en la imagen, de modo que es ampliamente utilizado en aplicaciones de robótica móvil [17].

El descriptor Gist (*Gabor-based Image Steepness*) se basa en el análisis de las respuestas a distintas frecuencias y orientaciones de los filtros de Gabor aplicados a una imagen. Estos filtros capturan características espaciales y texturales de la escena. El descriptor Gist se obtiene calculando las medias de las respuestas de los filtros en cada región de la imagen. Proporciona una representación compacta y eficiente de la estructura global de la escena. Este descriptor también se puede emplear para tareas de localización [18] [19], u otras que requieran información visual como el reconocimiento de objetos [20].

Además de los descriptores holísticos analíticos, basados en cálculos matemáticos, como HOG y Gist, también es posible obtener descriptores holísticos a partir de capas intermedias de redes neuronales convolucionales (CNN por sus siglas en inglés). Estas capas intermedias

calculan representaciones aprendidas de alto nivel de las imágenes. Al pasar una imagen a través de una CNN previamente entrenada, las capas de características extraen información semántica y abstracta que puede ser utilizada como descriptor holístico, una técnica empleada recientemente [21].

Una vez que se han extraído los descriptores holísticos, se pueden utilizar para resolver diferentes problemas relacionados con la tarea de localización, como la comparación de escenas, la estimación de posición o la detección de cambios en el entorno. Los descriptores se comparan con una base de datos de referencia que contiene información previamente registrada del entorno. Mediante técnicas de emparejamiento o clasificación, se busca encontrar la correspondencia más cercana entre los descriptores de la escena actual y los descriptores de referencia, lo que permite determinar la posición o la similitud con respecto a ubicaciones conocidas.

1.2. Inteligencia Artificial

La Inteligencia Artificial (IA) ha emergido como una tecnología con un impacto significativo en diversos campos, incluida la robótica móvil. La IA proporciona a los robots móviles capacidades avanzadas de percepción, toma de decisiones y adaptabilidad, lo que los hace más autónomos, eficientes y versátiles en sus tareas.

En el contexto de la robótica móvil, la IA se aplica en varias áreas, como el reconocimiento de objetos, el procesamiento del lenguaje natural, el aprendizaje automático y la planificación de rutas. Estas aplicaciones permiten a los robots móviles comprender y adaptarse al entorno en tiempo real, interactuar con los usuarios de manera natural y realizar tareas complejas de manera autónoma.

El aprendizaje automático, una rama de la IA, ha sido especialmente relevante en la robótica móvil. Los algoritmos de aprendizaje automático, como las redes neuronales [22] y los algoritmos de aprendizaje profundo [23], permiten a los robots móviles adquirir conocimiento y habilidades a través de la experiencia y mejorar su rendimiento con el tiempo. Esto facilita el desarrollo de sistemas de localización más precisos, sistemas de navegación autónoma más robustos y sistemas de interacción con el entorno y los humanos más naturales [24].

Una aplicación de la IA en la robótica móvil está en la planificación de rutas y la toma de decisiones [25]. Los algoritmos de planificación basados en IA permiten a los robots móviles encontrar la ruta óptima para navegar en entornos desconocidos o dinámicos, evitando obstáculos y optimizando criterios como la eficiencia energética o el tiempo de llegada. Además, los sistemas de toma de decisiones basados en IA pueden permitir a los robots móviles adaptarse a situaciones imprevistas y tomar decisiones inteligentes en tiempo real.

Otra área en la que la IA ha demostrado un gran potencial es en la percepción del entorno. Los algoritmos de visión por computadora y procesamiento de imágenes basados en IA permiten a los robots móviles reconocer y clasificar objetos, identificar obstáculos y to-

mar decisiones basadas en la información visual capturada por cámaras y otros sensores. Esto es especialmente relevante en entornos complejos y dinámicos donde los robots móviles deben adaptarse a cambios inesperados y tomar decisiones en tiempo real. En esta línea, el uso de Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) en particular ha generado gran interés. Las CNN son un tipo de red neuronal profunda que han demostrado un rendimiento sobresaliente en la tarea de reconocimiento de patrones en imágenes.

Una de las principales ventajas de emplear CNN en robótica móvil es su capacidad para extraer características visuales de alto nivel de las imágenes, permitiendo al robot comprender y analizar el entorno de manera más sofisticada. A través del entrenamiento con grandes conjuntos de datos, las CNN aprenden a reconocer y capturar patrones y características relevantes en las imágenes, lo que facilita la detección de objetos, la segmentación de escenas y la comprensión de la estructura espacial, además de la obtención de descriptores holísticos, como se mencionó anteriormente.

1.3. Motivación

El presente proyecto surge de la creciente necesidad de desarrollar sistemas de localización más precisos y robustos para robots móviles en entornos complejos y dinámicos. La localización es una tarea fundamental en la robótica móvil, ya que permite al robot conocer su posición y orientación en relación con el entorno circundante.

Aunque existen diversos enfoques y técnicas para la localización de robots móviles, se ha observado que los descriptores holísticos y el uso de inteligencia artificial presentan un gran potencial para mejorar la precisión y robustez de estos sistemas. Los descriptores holísticos capturan características globales de la escena, permitiendo una representación más completa y precisa del entorno. Además, la inteligencia artificial brinda herramientas avanzadas para el procesamiento de datos y la toma de decisiones, lo que puede mejorar significativamente la capacidad de los robots móviles para localizarse con precisión en entornos desafiantes.

Siguiendo esta línea de investigación, el objetivo principal de este proyecto es desarrollar algoritmos de localización probabilísticos y jerárquicos basados en descriptores holísticos y aprovechar el potencial de la inteligencia artificial para mejorar la precisión y robustez de la localización en robots móviles. Se busca aprovechar los avances recientes en técnicas de percepción visual, aprendizaje automático y procesamiento de datos para obtener resultados más precisos y confiables en la tarea de localización.

El enfoque propuesto en este proyecto presenta novedades importantes en la forma de emplear los descriptores holísticos y la inteligencia artificial. Se pretende desarrollar un sistema de localización que sea capaz de adaptarse a diferentes entornos y condiciones de iluminación, superando los desafíos asociados con la presencia de objetos superpuestos, materiales diversos y cambios en la iluminación. Además, se buscará integrar técnicas de aprendizaje automático para mejorar la capacidad de generalización y adaptabilidad del sistema.

Para lograr estos objetivos, se llevarán a cabo experimentos exhaustivos utilizando conjuntos de datos reales, se evaluarán diferentes parámetros que intervienen en el proceso estudiado y se compararán los resultados obtenidos con otros enfoques existentes en la literatura. Se espera que este trabajo contribuya al avance en el campo de la localización en robótica móvil y abra nuevas oportunidades para la aplicación de robots móviles en entornos más desafiantes y realistas.

1.4. Estructura de la memoria

Tras realizar una introducción del trabajo propuesto, el resto de la presente memoria se estructura de la siguiente manera:

En el capítulo 2 se presentará una revisión de los trabajos realizados durante los últimos años relacionados con la temática de la robótica móvil con información visual e inteligencia artificial

En el capítulo 3 se detallará la metodología empleada para el desarrollo de algoritmos de localización que se presentan en este trabajo

En el capítulo 4 se expondrán los resultados obtenidos de una serie de experimentos realizados con dichos algoritmos.

Finalmente, en el capítulo 5, se plantearán las conclusiones alcanzadas tras el desarrollo de este trabajo, así como las posibles líneas de trabajo futuras.

2 Estado del arte

En este capítulo, se revisarán algunos de los últimos avances en robótica móvil, así como los diversos métodos y técnicas empleadas para las tareas de navegación, mapeo y localización, con especial énfasis en esta última. Además, se estudiarán también trabajos basados en el reciente desarrollo de la Inteligencia Artificial, y cómo difieren estas técnicas frente a los métodos clásicos, presentando nuevos desafíos y oportunidades.

2.1. Localización jerárquica

Una de las técnicas más ampliamente utilizadas en robótica móvil es la localización basada en descriptores de apariencia visual. Esta técnica permite que los robots se ubiquen en un entorno desconocido utilizando la información visual que captan a través de sus cámaras.

Hay dos enfoques claramente diferenciados a la hora de aplicar esta técnica: por un lado, la utilización de descriptores de apariencia local, basados en extraer puntos o regiones características de la escena y obtener un vector de información para cada objeto seleccionado. Un ejemplo de uso sería el propuesto por Murillo et al. [18], en el cual se emplea el descriptor SURF (por sus siglas en inglés *Speeded Up Robust Features*). Este método detecta puntos de interés buscando mínimos locales en la matriz Hessiana y posteriormente crea una región circular alrededor de cada punto de interés para asignar una orientación única a cada uno y lograr invariancia a rotaciones de la imagen. Gracias a un algoritmo de supresión de no máximos eficiente, permite una velocidad de emparejamiento rápida.

Por otro lado, se puede optar por emplear descriptores de apariencia global, también llamados descriptores holísticos, que generan un único vector de información de la escena completa, como en el trabajo presentado por Céspedes [26], en el que se prueban algoritmos de localización basados en descriptores holísticos con escenas bajo condiciones de iluminación variables. No es el único trabajo propuesto en esta línea, también es notable el trabajo de Amorós et al. [27] en el que se emplean descriptores holísticos obtenidos a partir de imágenes omnidireccionales para resolver la localización de un robot móvil.

Este último planteamiento ha ganado más tracción dado que el tener un único descriptor simplifica los algoritmos de localización, que normalmente consisten en realizar una comparación entre las imágenes capturadas durante el proceso de localización y las imágenes almacenadas durante el proceso de mapeo en una base de datos. Este se puede aplicar directamente en entornos estructurados, obteniendo buenos resultados con tiempos de cómputo relativamente bajos.

A su vez, los descriptores de apariencia global se pueden subdividir en dos clases según la estrategia que emplean para ser calculados. En primer lugar están los descriptores holísticos basados en métodos analíticos, que consisten en aplicar alguna herramienta matemática para obtener un vector con información representativa de la imagen. Un ejemplo de uso lo podemos encontrar en Román et al. [28], donde se aplican descriptores holísticos como *HOG* y *gist* para llevar a cabo las tareas de mapeo y localización utilizando robots móviles; y en segundo lugar están los descriptores holísticos basados en *deep learning*, desarrollados gracias a los avances en inteligencia artificial y cuyo uso se está extendiendo rápidamente en los últimos años, como en el trabajo de Cebollada et al. [29] en el que se emplea como descriptor holístico la información contenida en una de las capas intermedias de una CNN encargada en principio de realizar una localización jerárquica.

Por otro lado, existen dos propuestas principales para llevar a cabo la tarea de localización mediante el uso de descriptores visuales. El primero consiste en realizar una localización *batch* (también conocido como algoritmo de búsqueda del vecino más cercano), esto es, mediante comparación directa entre el descriptor obtenido de la imagen capturada durante el proceso de localización y los descriptores calculados a partir de las imágenes capturadas durante el proceso de mapeo. La segunda propuesta de localización se conoce como localización jerárquica y consiste en construir un mapa jerárquico compuesto por diferentes capas de información. En los niveles superiores se recopila información más compacta que permite una localización más rápida sacrificando precisión, mientras que los niveles inferiores poseen información más detallada que permite afinar la posición exacta del robot en un área específica. Con este método, se consiguen resultados con un tiempo de cómputo menor y una precisión similar a los resultados que se obtienen mediante el método de localización *batch*. Un ejemplo de uso de localización jerárquica es el trabajo presentado por Cebollada et al. [30], donde se lleva a cabo un estudio sobre el uso de localización jerárquica con descriptores holísticos bajo cambios de iluminación.

2.2. Localización probabilística

Todos los sensores que posee un robot ofrecen medidas contaminadas por un cierto ruido que puede depender tanto del propio sensor y sus componentes, como de factores externos del entorno de trabajo del robot, de modo que estas medidas presentan una cierta incertidumbre que hay que tener en cuenta en la implementación de los algoritmos de localización.

Así surgen los métodos de localización probabilística, cuya principal utilidad reside en la capacidad de representar la incertidumbre asociada a la estimación de la posición del robot. A diferencia de los métodos no probabilísticos, que suelen proporcionar una única estimación de la posición del robot, los métodos probabilísticos proporcionan una distribución de probabilidad que representa la incertidumbre en la estimación.

Además, los métodos probabilísticos permiten la integración de información de múltiples fuentes, lo que puede mejorar la precisión de la estimación de la posición del robot. Por

ejemplo, en el caso de la localización basada en descriptores de apariencia visual, se puede utilizar la información visual junto con información de otros sensores, como los sensores de odometría, para mejorar la precisión de la estimación de la posición del robot, tal y como se propone en el trabajo propuesto por Valiente et al. [31].

El filtro de Bayes es un método que utiliza probabilidades para representar la incertidumbre en la posición del robot. Este método utiliza la información sensorial para actualizar la probabilidad a posteriori de la posición del robot, utilizando la ley de Bayes. Uno de los algoritmos más populares basados en el filtro de Bayes es el algoritmo de localización de Monte Carlo (MCL) [32], también conocido como filtro de partículas. La simulación de Monte Carlo es una técnica que se utiliza para simular una distribución de probabilidades para la posición del robot. Este método genera muestras aleatorias de la distribución de probabilidad y utiliza estas muestras para estimar la posición del robot. El trabajo de Xu et al. [33] aborda el problema de la localización visual en robots móviles mediante el uso de una técnica de reconocimiento de lugares visuales (VPR, por sus siglas en inglés) basada en la probabilidad. Los autores proponen un marco de trabajo jerárquico de localización basado en la integración de múltiples hipótesis de localización generadas por el VPR, lo que les permite mejorar la precisión y la eficiencia de la localización. La técnica propuesta se basa en la extracción de descriptores de apariencia visual de las imágenes y en la comparación de estos descriptores con una base de datos de imágenes previamente almacenadas en el robot. Además se combina con una técnica de muestreo de Monte Carlo para la estimación de la incertidumbre en la localización y un enfoque de fusión de información para integrar diferentes fuentes de información.

El filtro de Kalman es otro método popular de localización probabilística que utiliza un modelo matemático para estimar la posición del robot en función de la información sensorial. Este método utiliza un modelo dinámico para predecir la posición futura del robot y la información sensorial para corregir la estimación. Hua et al. [34] aborda el problema de la imprecisión en la localización y mapeo de robots móviles equipados con un solo sensor, debido a factores como objetos superpuestos, materiales diferentes e iluminación desigual en entornos interiores. Para solucionar este problema, se propone un enfoque de fusión de datos de múltiples sensores (LiDAR, cámara de profundidad, IMU y encoder) para la localización y mapeo de robots móviles en interiores. En particular, se utiliza el algoritmo de filtro de Kalman extendido para fusionar los datos del encoder con la unidad de sensores inerciales, lo que reduce el error de deriva y mejora la precisión de la localización del robot. Además, se integra información visual adicional en los datos de LiDAR 2D mediante el algoritmo de proximidad de región, lo que compensa la falta de percepción espacial en el mapeo de LiDAR de una sola línea y mejora la integridad espacial del mapeo del robot.

2.3. Inteligencia artificial

Entre las técnicas basadas en inteligencia artificial aplicadas a la robótica móvil, se destaca el uso de CNN, que se han utilizado ampliamente en visión por computador y robótica móvil para extraer características descriptivas de imágenes [35], y para resolver problemas de clasificación [36] y reconocimiento de espacios[37].

En particular, este trabajo se centra en explorar el uso de CNN para la tarea de localización gruesa del robot, por ejemplo para determinar a qué habitación o área pertenece la foto tomada antes de afinar la posición exacta dentro de la misma.

Como se mencionó en apartados anteriores, existen descriptores holísticos basados en inteligencia artificial. Este trabajo propone la obtención de descriptores de apariencia global a partir de capas intermedias de una red convolucional. Para ello, se entrena la red con un conjunto de datos de imágenes etiquetadas previamente, donde cada imagen tiene asociada una etiqueta que indica la clase a la que pertenece. Durante el entrenamiento, la CNN aprende a extraer características descriptivas de las imágenes que son relevantes para la clasificación. Ambas funcionalidades pueden ser desempeñadas por la misma red [38] [30]. Una vez la red ha sido entrenada, al introducir una imagen nunca antes vista en la red, esta será capaz de clasificarla de manera correcta. Además, se generará un descriptor holístico lo suficientemente robusto para poder establecer una comparativa de similitud con descriptores de otras imágenes.

Por otro lado, cabe destacar la necesidad de grandes cantidades de datos etiquetados para entrenar y ajustar los modelos. Esto se debe a que estas técnicas se basan en el aprendizaje a partir de ejemplos, donde el algoritmo debe analizar un gran número de ejemplos etiquetados para poder aprender a realizar una tarea determinada.

Cuanto más grandes sean los conjuntos de datos, más generalizable será el modelo, lo que significa que será capaz de manejar mejor situaciones nuevas y desconocidas. También es importante que los datos de entrenamiento sean representativos del mundo real en el que se espera que opere el robot. Por eso, la disponibilidad de grandes cantidades de datos es fundamental, pero es común disponer de un conjunto de datos relativamente limitado, por ello se recurre a la técnica de *data augmentation* [39], que consiste en crear variantes de las muestras originales que conserven las características relevantes del problema, pero que presenten pequeñas variaciones que permitan a la red aprender a reconocer patrones en una variedad más amplia de datos. Por ejemplo, en el caso de imágenes, se pueden aplicar transformaciones como rotaciones, cambios de escala, recortes aleatorios, ajustes de brillo y contraste.

Estas transformaciones se aplican aleatoriamente a las muestras originales durante el entrenamiento, lo que resulta en un conjunto más amplio y diverso de datos de entrenamiento. Además, la aplicación de *data augmentation* puede ayudar a prevenir el sobreajuste, ya que obliga a la red a generalizar mejor y no depender demasiado de características específicas de las muestras originales, tal y como se expone en el trabajo de Wan et al. [40] en el que se

aplica esta técnica para entrenar una CNN más robusta para resolver la localización de un robot móvil. También se puede aplicar para tareas de detección de objetos, como se propone en el trabajo de Zhang et al. [41], en el que se logran detectar satisfactoriamente pequeños objetos en movimiento, a pesar de contar con una cantidad reducida de datos de partida.

Otra técnica de especial mención es la de *transfer learning*, que, en esencia, se aprovecha el conocimiento adquirido por un modelo previamente entrenado en una tarea para mejorar el rendimiento de un modelo en una tarea relacionada pero distinta. Esta técnica se ha convertido en una herramienta valiosa para el aprendizaje eficiente y rápido en entornos de robótica móvil. Por ejemplo, en lugar de entrenar un modelo desde cero para una tarea específica, se puede utilizar un modelo pre-entrenado en una tarea relacionada, como el reconocimiento de objetos, y luego ajustarlo con un conjunto de datos más pequeño y específico de la tarea en cuestión. Así se puede reducir significativamente el costo computacional y de tiempo asociado con el entrenamiento de modelos desde cero, especialmente cuando se trabaja con grandes conjuntos de datos o tareas complejas. Además, el uso de un modelo pre-entrenado en una tarea relacionada también puede ayudar a evitar problemas de sobreajuste y a mejorar el rendimiento general del modelo. Esta técnica ha demostrado, con trabajos como el Chiba et al [42], que permite reducir la cantidad de datos necesarios y tiempo empleado para el entrenamiento de redes neuronales. Además, esta técnica se puede aplicar para el entrenamiento de todo tipo de redes neuronales propuestas en otros trabajos [43].

3 Metodologías

En este capítulo se van a describir los algoritmos de localización desarrollados, así como el trabajo y material en que se basan.

Los algoritmos que se van a presentar están basados en la obtención de información visual del entorno en el que se encuentra el robot. Para ello se emplean cámaras omnidireccionales que permiten captar en una imagen todo el entorno circundante. A partir de estas imágenes se obtienen descriptores holísticos, que contienen la información necesaria para desarrollar estos métodos.

Para comprobar el funcionamiento de los algoritmos, se realizarán una serie de experimentos con imágenes de la base de datos COLD (acrónimo del nombre completo *COsy Localization Database*) [44], elaborada por investigadores de la Universidad de Friburgo, con la ayuda de un robot móvil y cámaras omnidireccionales.

El robot móvil empleado cuenta con sensores SICK tanto láser como de odometría, que son notoriamente fiables, y al combinar la información de ambos, es posible obtener el *ground truth*, es decir, la información exacta de posición en la que se ha tomado cada imagen, esta información se usará más adelante para evaluar el error de localización.

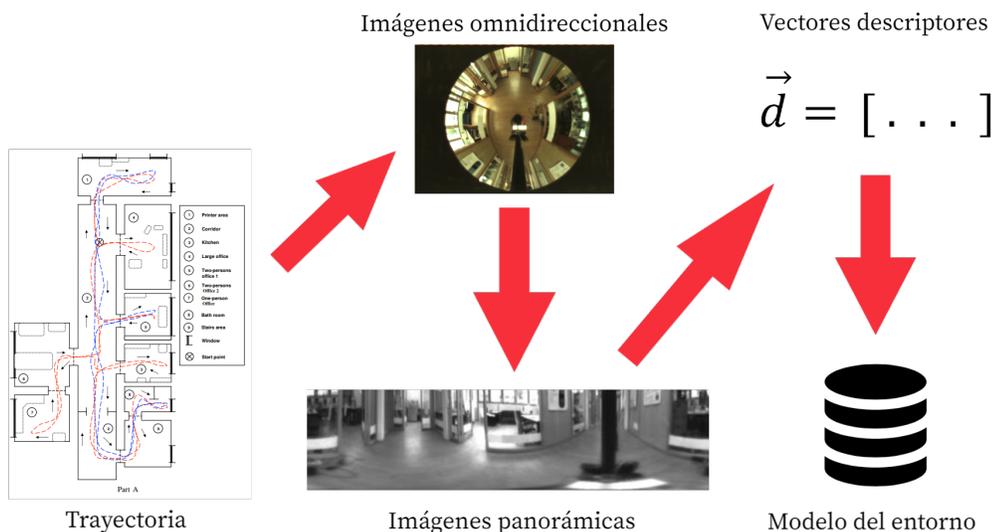


Figura 3-1: Proceso de mapeado.

Las imágenes omnidireccionales tomadas son imágenes con información de color RGB con una resolución de 470x470 píxeles. Estas imágenes son procesadas para ser transformadas en imágenes panorámicas, con unas dimensiones de 512x128 píxeles. A partir de estas últimas, se obtiene un descriptor holístico, que se almacena en una base de datos que conformará el modelo del entorno, tal y como se ilustra en la figura 3-1.

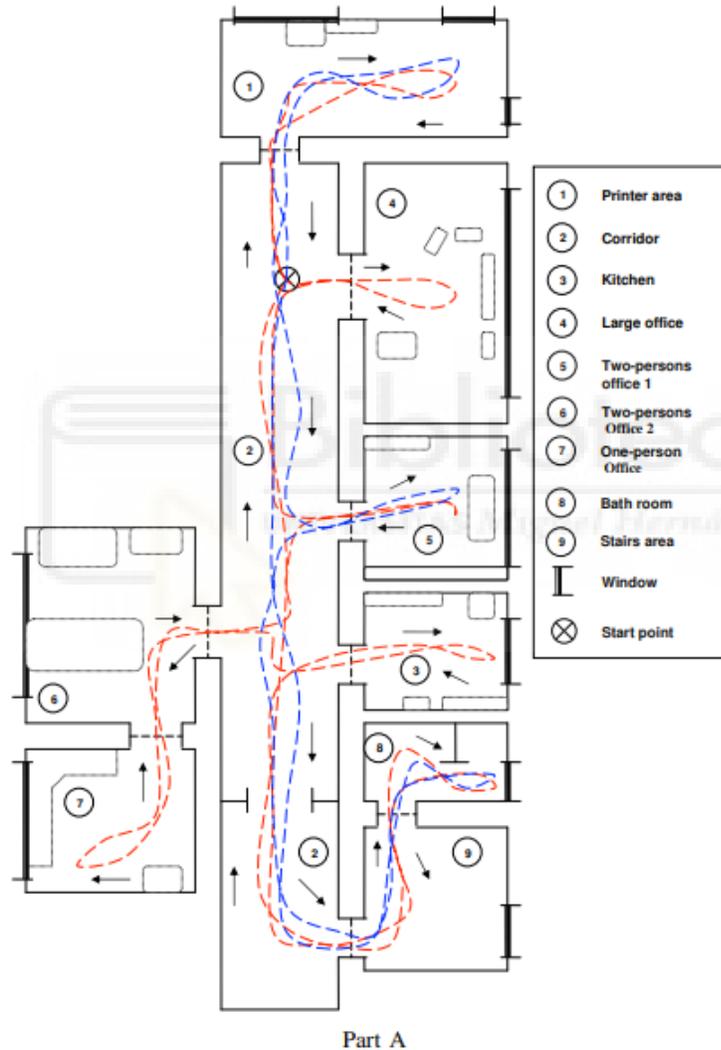


Figura 3-2: Mapa Friburgo. Fuente [45]

Las imágenes se toman espaciadas una cierta distancia mientras el robot sigue una trayectoria predefinida, tal y como se ven en la figura 3-2. En la base de datos completa, hay disponibles conjuntos con diferentes condiciones de iluminación (nublado, soleado, noche), así como varios conjuntos del recorrido realizado en diferentes ocasiones con condiciones de iluminación similares. Para este trabajo, se usa un conjunto de imágenes con iluminación nublada, muestreado para coger solo 1 de cada 5 imágenes, para elaborar el modelo del en-

torno. Se utilizará, por otro lado, otro de los conjuntos también con iluminación nublada, como imágenes de test para comprobar cómo funcionan los algoritmos.

A continuación se presentan todos los algoritmos desarrollados en este trabajo.

3.1. Localización batch

El método consiste en la búsqueda del vecino más cercano, es decir, comparar cada imagen de test con todas las imágenes obtenidas en el proceso de mapeo y determinar así la más parecida desde el punto de vista visual, que será seleccionada como la posición en que se encuentra el robot.

En primer lugar, se elabora el modelo de entorno (proceso de mapeo), para lo cual se toman todas las imágenes de entrenamiento y se obtiene el descriptor holístico de cada una de ellas, así como sus respectivas coordenadas, que se se consideran información conocida del entorno (a diferencia de las coordenadas del test, el *ground truth*, que se asumirá desconocido a priori y que únicamente se utilizará para llevar a cabo la evaluación de los métodos de localización).

En este método se toman todas las imágenes de test, y una a una se obtiene el descriptor holístico correspondiente, que tiene que ser el mismo que el modelo con el que va a ser comparado. A continuación se calcula la distancia entre el descriptor de test y todos los descriptores del entorno. Se puede evaluar la distancia euclidea, coseno o manhattann.

Así, para cada imagen de test se busca la menor distancia con las imagenes del modelo y la que obtenga la menor distancia se considera la posición del robot. Entonces, se compara con el dato de *ground truth* correspondiente para obtener el error de localización. El funcionamiento de este algoritmo se puede resumir con el diagrama de la figura 3-3

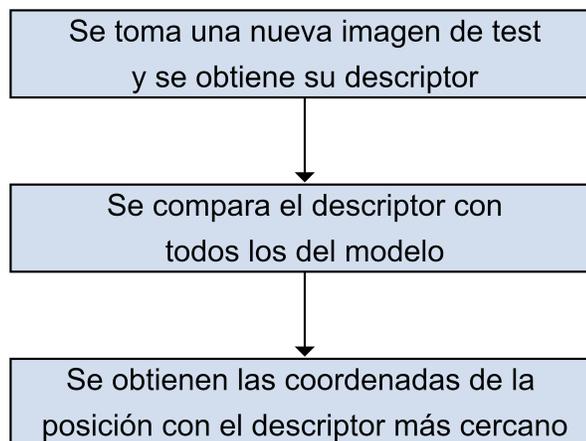


Figura 3-3: Diagrama de flujo del algoritmo de búsqueda del vecino más cercano.

3.2. Localización jerárquica

En este método se pretende reducir el número de comparaciones respecto al anterior haciendo una búsqueda jerárquica, que consiste en una localización gruesa que permite discernir la zona aproximada en la que se encuentra el robot, seguida de una localización fina, en la que se determina exactamente la posición que ocupa en el mapa. Para ello, es necesario agrupar de algún modo el conjunto de imágenes de entrenamiento.

Dicho agrupamiento se realiza empleando la técnica de *clustering* espectral [46], que toma todo el conjunto de descriptores de las imágenes de mapeo, y un número predefinido de clusters; y le asigna a cada uno una etiqueta, de modo que quedan organizados en grupos.

Una vez realizado el agrupamiento, se va a tomar un vector representativo de cada *cluster*, que es la media de todos los vectores que tengan la misma etiqueta.

Entonces, para cada imagen de test, se obtiene su vector descriptor que primeramente se compara con los representativos para determinar a qué *cluster* debería pertenecer, es decir, se realiza la localización gruesa; y a continuación se compara este mismo descriptor de test solamente con aquellos del modelo que tengan la misma etiqueta para buscar aquel con el que tenga menor diferencia. De este modo al final se obtiene la posición que se supone que ocupa el robot en el modelo, que sería la localización fina. Este último paso se realiza de forma similar al método anterior, pero con un número reducido de comparaciones. Todo esto se recoge en el diagrama de la figura 3-4.

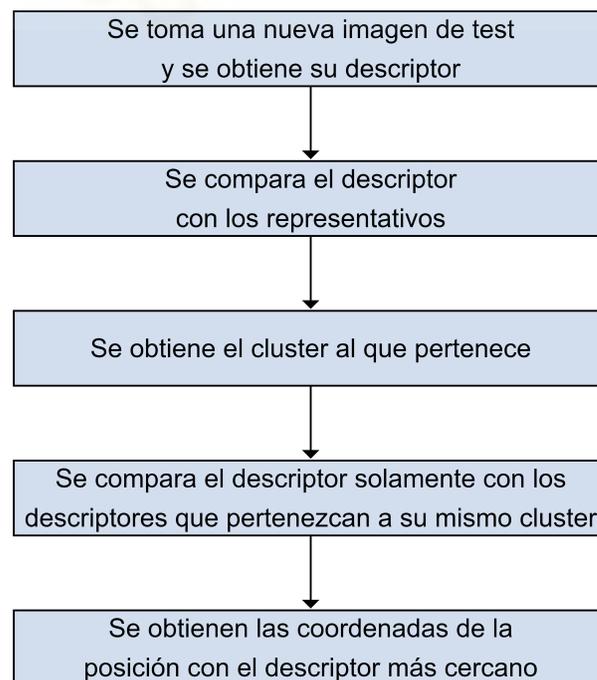


Figura 3-4: Diagrama de flujo del algoritmo jerárquico.

3.3. Localización probabilística con información métrica de baja confianza

Este es el primero de los métodos probabilísticos que se presentan en este trabajo. Los métodos anteriores siempre tienen como objetivo localizar al robot en una de las posiciones del mapa. Sin embargo, lo más probable es que durante el funcionamiento real, el robot intente localizarse cuando no se encuentra en una posición exacta del mapa, tanto si es en una posición intermedia como en una fuera de la trayectoria. Por este motivo, se propone asignar valores de probabilidades a cada posición del mapa, recogidos en una función de densidad de probabilidad (fdp, de aquí en adelante).

La ventaja de emplear un método así es que se puede tener en cuenta la incertidumbre y ruidos por los que están afectadas las medidas de los sensores. Además, también se tiene en cuenta un movimiento del robot más acorde con la realidad, que será de forma continua siguiendo la trayectoria predefinida, en lugar de pasar de una posición a otra del modelo de forma discreta, como se asumía anteriormente. Así, también se puede estimar una posición final que corresponda exactamente con una posición del mapa tomando los respectivos pesos de cada posición.

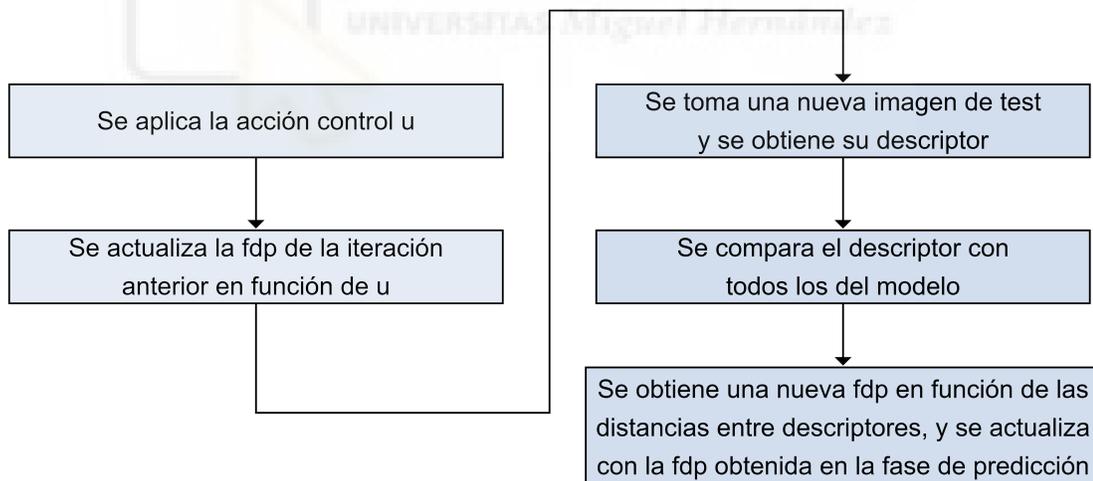


Figura 3-5: Diagrama de flujo del algoritmo probabilístico con información métrica de baja confianza.

Para este método la hipótesis de partida, es que se disponen datos de odometría durante el desplazamiento del robot. Estas medidas solo tienen en cuenta cuánto giran las ruedas de los motores, de modo que es muy fácil que se cometan errores en la localización al usar solamente esta información, debido a causas como holguras en los componentes mecánicos o patinaje de las ruedas, entre otras. Por este motivo, la información de movimiento solo se usará para realizar una primera estimación sobre la posible posición del robot, que luego habrá que complementar con la información visual.

Este método se basa en el filtro de Bayes, ampliamente utilizado en robótica móvil para la estimación del estado del robot. De forma probabilista como se ha descrito, se añade información sobre las nuevas observaciones a partir de la hipótesis de la iteración anterior. Se estructura en dos fases: la de predicción, en la que se aplica el modelo movimiento, seguida de la fase de corrección en la que se aplica el modelo de observación.

En la fase de predicción, al aplicar el modelo de movimiento, se propaga la fdp actual según la acción de control $\vec{u} = [\vec{v}_x, \vec{v}_y]$, que encapsula las velocidades lineales y angulares que es capaz de tomar el robot, con su incertidumbre asociada. Para cada posición del modelo, se calcula la probabilidad de caer en dicha posición partiendo de cada una de todas las posiciones del modelo con el comando u , y se multiplica por su probabilidad asociada de la fdp obtenida en la iteración anterior; con lo cual se obtiene la probabilidad de estar en dicha posición, condicionada a la fdp anterior y la acción de control actual.

En la etapa de corrección se aplica el modelo de observación, que consiste en hacer una comparación del descriptor de la imagen de test con todas las imágenes del modelo, de forma similar a como se hacía en el método batch, con la salvedad de que para este método no se selecciona solamente la menor distancia, sino que se obtiene una fdp en la que las probabilidades son inversamente proporcionales a las distancias entre los descriptores. Finalmente se compara esta observación realizada con la predicción del modelo de movimiento, manteniendo la consistencia entre ambas fuentes de información. Para ello, se multiplica esta última fdp con la resultante de la etapa de predicción para obtener la fdp de la iteración que integra toda la información disponible.

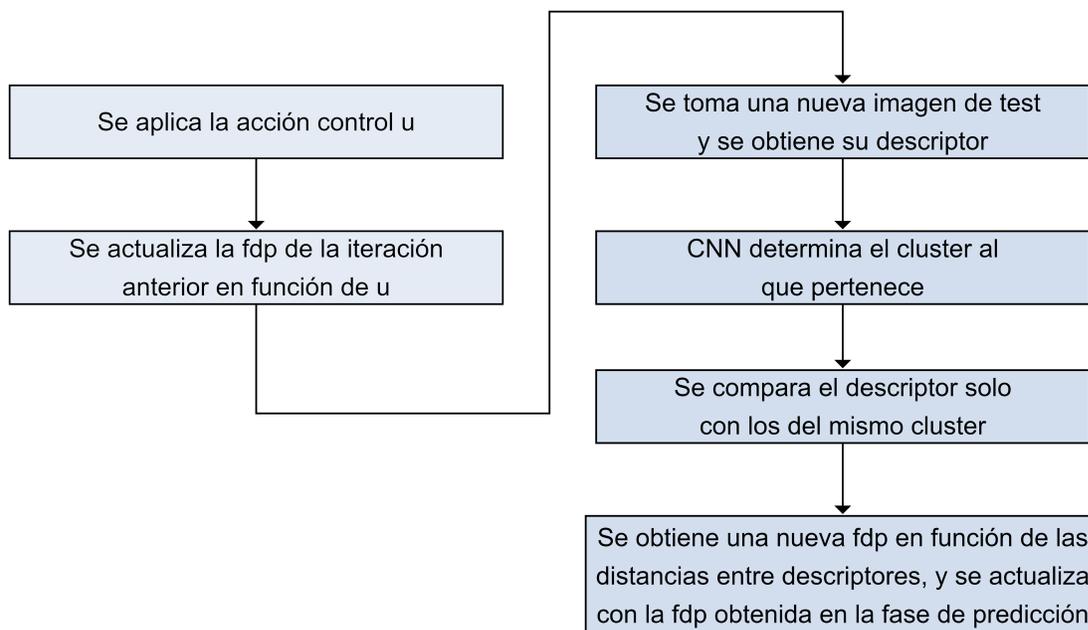


Figura 3-6: Diagrama de flujo del algoritmo jerárquico probabilístico.

Para poner en funcionamiento el método, se parte de la primera posición de la trayectoria

y se simula un movimiento nulo, pero con cierta incertidumbre. De este modo se obtiene una fdp que sirve como punto de partida para las siguientes iteraciones. A partir de ahí, en cada iteración se aplican sucesivamente el modelo de movimiento y el de observación.

Para la estimación final de la posición, como se obtiene una fdp que representa la probabilidad de que el robot ocupe cada posición del entorno, se puede tomar la posición más probable como la estimación, o se puede hacer una media ponderada de las posiciones más probables para hacer una mejor estimación que no necesariamente se encuentre justamente en una de las posiciones del mapa. El proceso completo se esquematiza en la figura 3-5.

3.4. Localización jerárquica probabilística

Este método implementa el filtro de Bayes del mismo modo que el anterior, pero en esta ocasión, el modelo de observación se asemeja más al método jerárquico presentado anteriormente, en lugar del método batch.

Sin embargo, la implementación de este modelo de observación también es diferente a lo descrito anteriormente. Ahora se propone el uso de una CNN para realizar la clasificación en *clusters*, de forma análoga a la localización gruesa del método jerárquico, como se indica en la figura 3-6.

3.4.1. Entrenamiento CNN

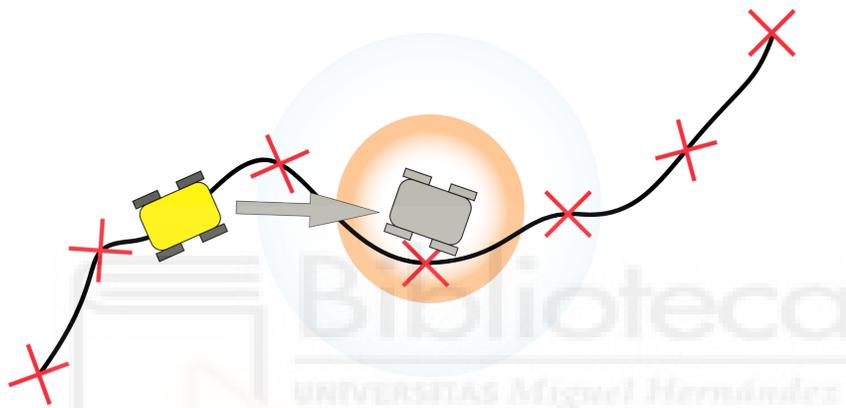
Se pretende entrenar una CNN que sea capaz de clasificar una imagen cualquiera tomada durante el recorrido según el agrupamiento en *clusters* que se había realizado anteriormente. Existen varias arquitecturas de redes que pueden solucionar este problema. En este trabajo se propone emplear redes tales como LeNet o AlexNet.

Sea cual sea la arquitectura escogida, estas redes han sido entrenadas y probadas para tareas particulares de clasificación, diferentes a la que se necesita en este trabajo. Por ello, es imprescindible reentrenar las redes para que den un resultado satisfactorio en esta tarea. Se realizará el entrenamiento con el conjunto de imágenes de mapeo, haciendo validación cruzada en cada iteración para evitar el sobreajuste. De este modo, se espera que la red sea capaz de clasificar de forma robusta las imágenes del conjunto de test, que tendrán más variabilidad debido a que se dispone de una mayor cantidad de ellas a lo largo de la trayectoria.

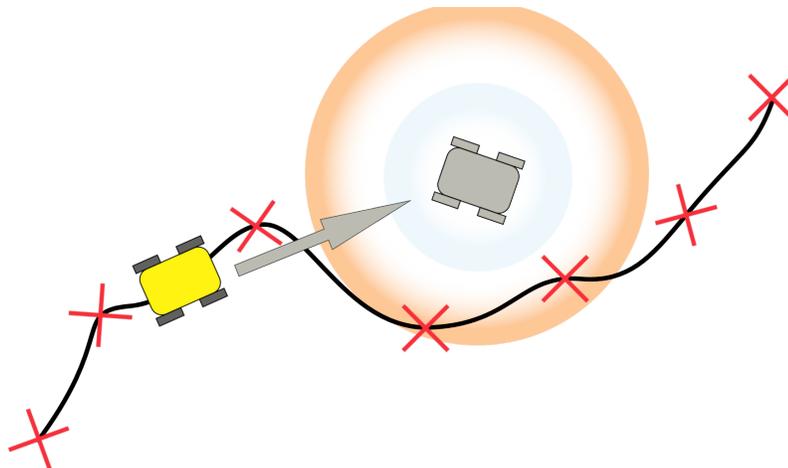
Por otro lado, también se pretende que la CNN sea capaz de obtener un descriptor holístico de cada imagen procesada. Debido a la estructura de funcionamiento de las CNN, la información de entrada, una imagen en este caso, se procesa de modo que se van extrayendo características importantes que las diferencien unas de otras. Esta cualidad se puede aprovechar, extrayendo la información de una capa intermedia de la red, que se puede usar como descriptor holístico.

3.5. Localización probabilística con información métrica de confianza

Para este método se parte de la hipótesis que que la información métrica para estimar la posición en la etapa de movimiento proviene de un sensor como GPS o LiDAR, que proporciona medidas mucho más fiables que la odometría que se había considerado en los métodos anteriores



- (a) Se toma el umbral más pequeño siempre que haya posiciones cercanas del mapa.



- (b) Se toman el umbral más grande en caso de que el primero no alcance ninguna posición del mapa.

Figura 3-7: Ejemplos del umbral dual de búsqueda.

De este modo, se puede emplear la información obtenida por el modelo de movimiento para reducir en gran medida el número de comprobaciones que se tienen que realizar en la

etapa de corrección. En primer lugar se toma la posición más probable de la distribución de probabilidades de la iteración anterior y, partiendo de ella, se aplica la acción de control que se obtiene comparando las coordenadas de *ground truth* de la imagen actual con la siguiente. Además, igualmente se aplica un cierto ruido gaussiano a esta acción de control para simular el funcionamiento de un sensor real. Luego, a partir de la posición alcanzada tras moverse, se seleccionan las posiciones del mapa que quedan lo suficientemente cerca, por debajo de un umbral previamente seleccionado. Como las posiciones del mapa siguen una cierta trayectoria y el ruido puede desviar la posición alcanzada en cualquier dirección, es posible que no sea posible seleccionar ninguna posición del mapa si el umbral considerado es demasiado pequeño; si se da este caso, se coge un segundo umbral más amplio para asegurarse de poder coger algunas posiciones del mapa, tal y como se ilustra en la figura 3-7.

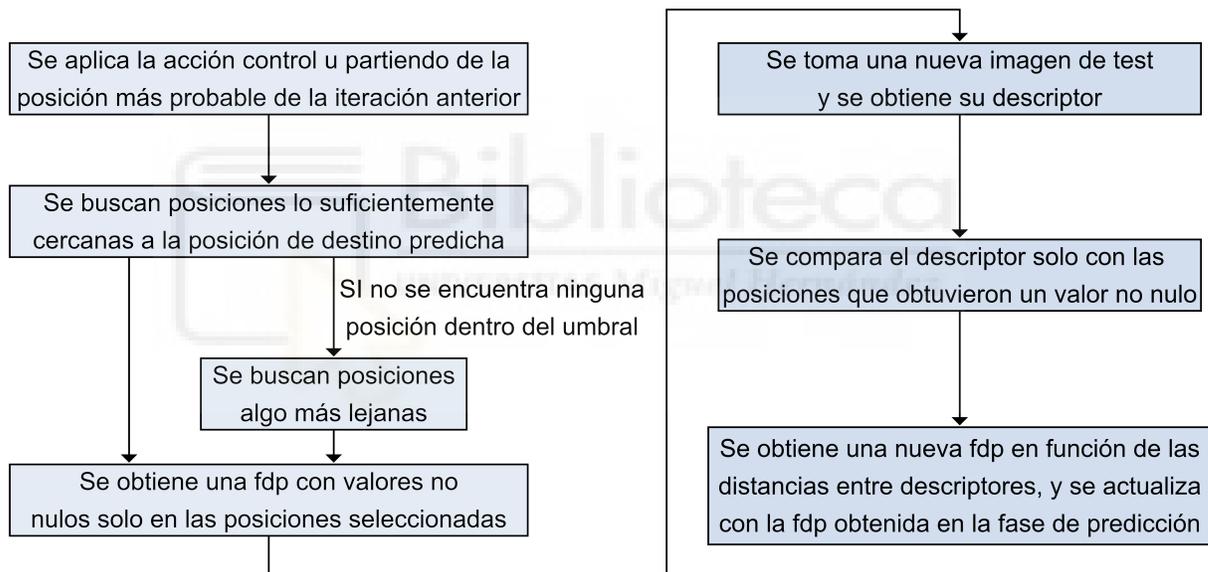


Figura 3-8: Diagrama de flujo del algoritmo con información métrica de confianza.

Una vez seleccionadas estas posiciones, se les asigna una probabilidad inversamente proporcional a la distancia a la posición en la que se cree que está el robot al moverse; mientras que todas las probabilidades que queden fuera del umbral tendrán una probabilidad de cero. Así, en cierto modo, esta primera etapa de predicción se asemejaría a la localización gruesa de los métodos jerárquicos.

Finalmente, en la etapa de corrección, se aplica el modelo de observación, que compara el vector descriptor de la imagen de test con el de las imágenes del mapa que han obtenido una probabilidad no nula en la etapa anterior; asignando a cada posición una probabilidad inversamente proporcional a la distancia entre los descriptors comparados, y condicionado a la respectiva probabilidad de la etapa anterior. De este modo, esta etapa se correspondería con la localización fina de los métodos jerárquicos, con la diferencia de que la búsqueda jerárquica se ha basado en información métrica en vez de en un proceso de *clustering* anterior.

Se resume el funcionamiento completo del algoritmo en la figura 3-8. Además, de este modo es posible controlar el número de comprobaciones a realizar simplemente variando los umbrales a considerar en la primera etapa.

3.6. Localización probabilística sin información métrica

En este último método propuesto en este trabajo, se considera como hipótesis de partida que no se dispone de GPS o LiDAR, y que no hay información de odometría o que dicha información presenta un alto nivel de ruido; de modo que no se puede utilizar la información métrica para la etapa de predicción. Por lo tanto se supone que en cada instante de muestreo, el robot ha podido moverse una distancia igual a la que podría moverse a su velocidad máxima en ese intervalo de tiempo, siguiendo estrictamente la trayectoria del mapa en cualquier sentido. De este modo, se parte de la posición más probable de la iteración anterior, y se considera que el robot pudiera acabar hasta N posiciones hacia adelante o hacia atrás en el mapa, asignando probabilidades no nulas a las posiciones que que estén dentro de esta ventana, como se puede ver en la figura 3-9

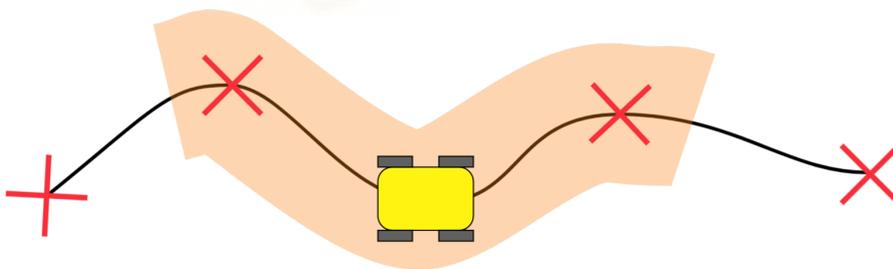
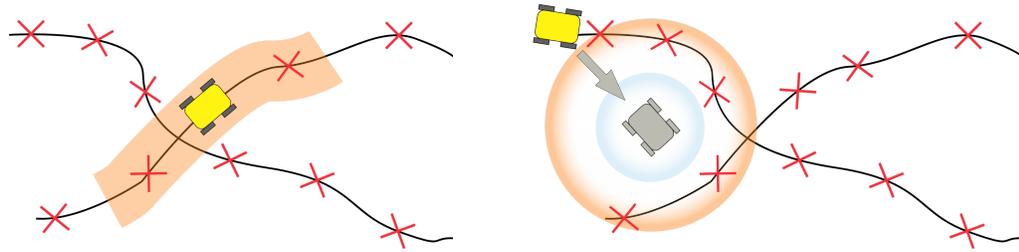


Figura 3-9: Ventana de búsqueda del método sin información métrica.

Así, se obtiene una distribución de probabilidades similar al método anterior, con la salvedad de que este método no puede considerar posiciones métricamente cercanas si está separadas en el orden de la trayectoria del mapa, lo cual supone una diferencia importante respecto al método anterior, tal y como se puede apreciar en la figura 3-10.

En cualquier caso, a continuación se aplica la etapa de corrección sobre las posiciones con probabilidades no nulas, de forma idéntica al método anterior, para obtener la distribución



- (a) Se toman las posiciones consecutivas aunque haya posiciones métricamente más cercanas.
- (b) El método anterior sigue tomando las posiciones más cercanas, independientemente de su lugar en la trayectoria.

Figura 3-10: Comparativa en caminos cruzados.

de probabilidades final en cada iteración. Todo esto queda reflejado en el diagrama de la figura 3-11.

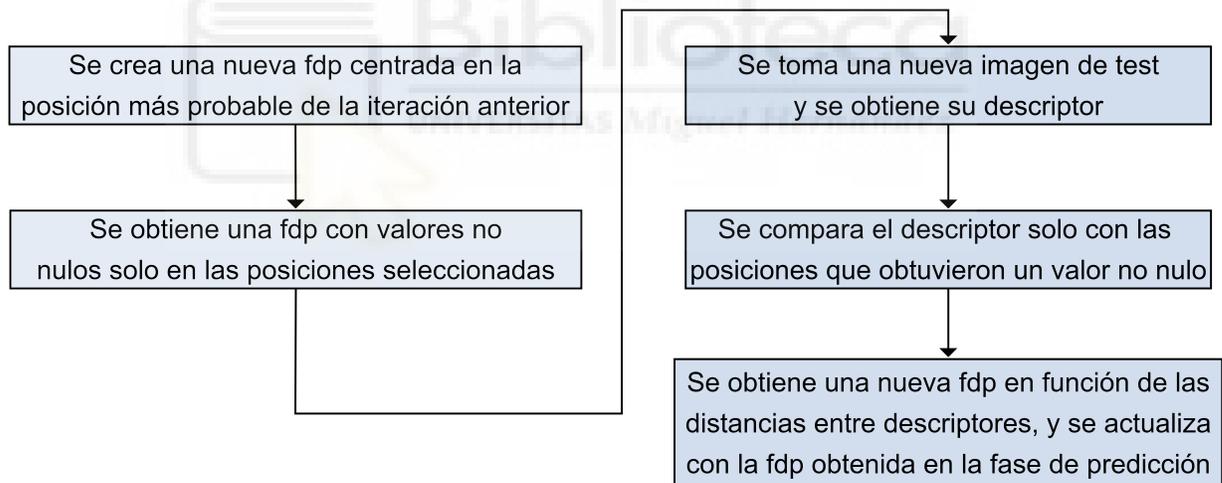


Figura 3-11: Diagrama de flujo del algoritmo sin información métrica.

4 Experimentos y resultados

En este capítulo, se van a detallar los experimentos realizados con los algoritmos descritos anteriormente. Para ello se va a simular el funcionamiento del robot en movimiento, siguiendo la trayectoria del conjunto de imágenes de testeo, tomando de forma sucesiva cada una de estas imágenes y obteniendo una estimación de su posición teniendo en cuenta las coordenadas de las imágenes del conjunto de mapeo, que es información conocida previa al experimento.

4.1. Experimentos previos

Como trabajo preliminar a ciertos experimentos sobre los algoritmos de localización, se propone emplear una CNN para hacer la clasificación de las imágenes tomadas, y previamente etiquetadas mediante el clustering espectral. De este modo, se podrá hacer la localización gruesa de los métodos jerárquicos a través de la CNN.

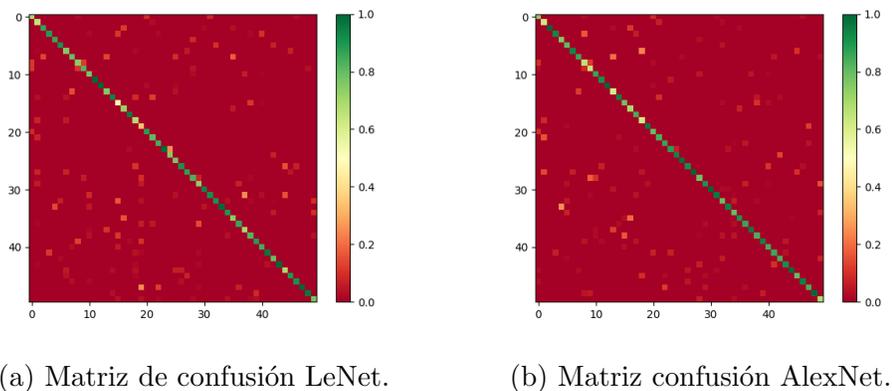


Figura 4-1: Matrices de confusión.

Se ha estudiado el uso de varias arquitecturas ya usadas en una amplia variedad de problemas, como LeNet [47] y AlexNet [48]. Estas redes han sido entrenadas para determinadas tareas de localización, pero es posible modificar las capas de entrada y de salida para adaptarlas a la clasificación a realizar en este caso, y posteriormente reentrenar las redes.

En particular, se va a modificar la capa de entrada de forma que tome las imágenes panorámicas en blanco y negro que se emplearon para el proceso de mapeo. La capa de

	LeNet	AlexNet
Micro Accuracy	0.7922	0.8458
Micro Precision	0.7922	0.8458
Micro Recall	0.7922	0.8458
Micro F1-score	0.7922	0.8458
Macro Accuracy	0.9917	0.9938
Macro Precision	0.8304	0.8549
Macro Recall	0.7952	0.8455
Macro F1-score	0.7903	0.8430
Weighted Accuracy	0.9913	0.9937
Weighted Precision	0.8335	0.8613
Weighted Recall	0.7922	0.8458
Weighted F1-score	0.7883	0.8468

Tabla 4-1: Métricas de entrenamiento CNN

salida se modifica para clasificar este conjunto de imágenes en 50 clases, según un clustering espectral que se ha realizado previamente. Este clustering se ha realizado obteniendo el descriptor gist de cada imagen, probando con distintos números de clusters hasta obtener el mejor resultado, optando finalmente por 50 clusters, tal y como se detallará en un apartado posterior, obteniendo el mapa segmentado que se observa en la figura 4-5.

Se realizará el entrenamiento con el conjunto de imágenes de mapeo, con 519 imágenes en total. Durante el entrenamiento, se hace uso de la técnica de *data augmentation*, que permite aplicar transformaciones tales como rotaciones y oclusiones aleatorias para aumentar la variabilidad y enriquecer este conjunto de datos. Asimismo, se hace también una validación cruzada, es decir, que en cada *epoch* del entrenamiento, una parte del conjunto de imágenes seleccionada al azar se emplea para validar los pesos entrenados.

Tras el entrenamiento, se evalúan una serie de métricas [49] que determinan la bondad de la red en la tarea de clasificación. Esta evaluación se hace con el conjunto de imágenes de testeo, 2595 en total, cuyas etiquetas reales se desconocen, por lo que se asume para cada una de ellas, que le corresponde la etiqueta de la imagen más cercana del conjunto de mapeo con el que se realizó el entrenamiento. De estas métricas, recogidas en la tabla 4-1, *Recall* se refiere al número de elementos identificados correctamente como positivos (es decir, que pertenece a una cierta clase) del total de positivos verdaderos; *Precision* es el número de elementos identificados correctamente como positivo del total de elementos identificados como positivos; *Accuracy* es el porcentaje total de elementos clasificados correctamente; y *F1-Score* combina *Precision* y *Recall* en una sola medida, es la media armónica de ambas. Además, estos resultados también se pueden visualizar en las matrices de confusión, en

las figuras 4-1a y 4-1b; donde las clasificaciones correctas deben caer en la diagonal de la matriz, ya que el eje horizontal representa la clase predicha y el eje vertical la clase real. Ambas redes obtienen un resultado bastante bueno, aunque AlexNet es ligeramente mejor para esta instancia de *cluster retrieval* que estamos realizando con las redes.

4.2. Experimentos de localización

4.2.1. Localización batch

En primer lugar, se realiza el experimento de localización por búsqueda del vecino más cercano. Se estudia el uso de diferentes descriptores holísticos, el descriptor analítico HOG, así como algunos obtenidos de capas intermedias de CNN, como las versiones reentrenadas de LeNet y AlexNet que se han presentado previamente; y también de VGG16 sin reentrenar, para evaluar el funcionamiento de distintas redes. Para cada descriptor, se elabora previamente el correspondiente modelo para poder comparar con los descriptores de testeo obtenidos durante el experimento.

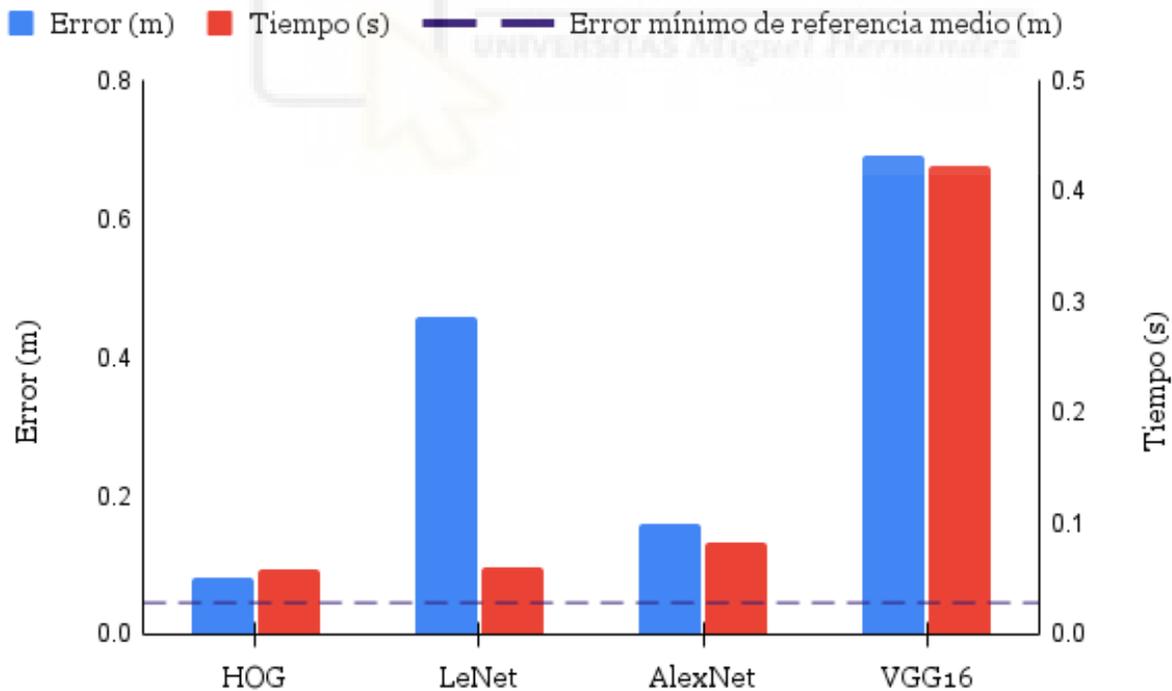


Figura 4-2: Resultados del método del vecino más cercano.

En la figura 4-2 se comparan los resultados de error de localización medio para cada uno de los diferentes descriptores con el valor mínimo de error de referencia medio, así como una comparación de los tiempos de cómputo de cada uno. Se observa cómo el descriptor

analítico HOG da el mejor resultado, pero los descriptores obtenidos de redes neuronales pueden llegar a obtener resultados comparables, en este caso el de AlexNet.

Por otro lado, se muestra en forma de histograma, para cada descriptor empleado, si se ha seleccionado la imagen más cercana del modelo del entorno, representada por el valor 0 de proximidad, o si se escogen imágenes más alejadas y con qué frecuencia ocurre esto, tal y como se observa en la figura 4-3. En particular se observa en la figura 4-3d cómo comete errores importantes seleccionando imágenes muy lejanas.

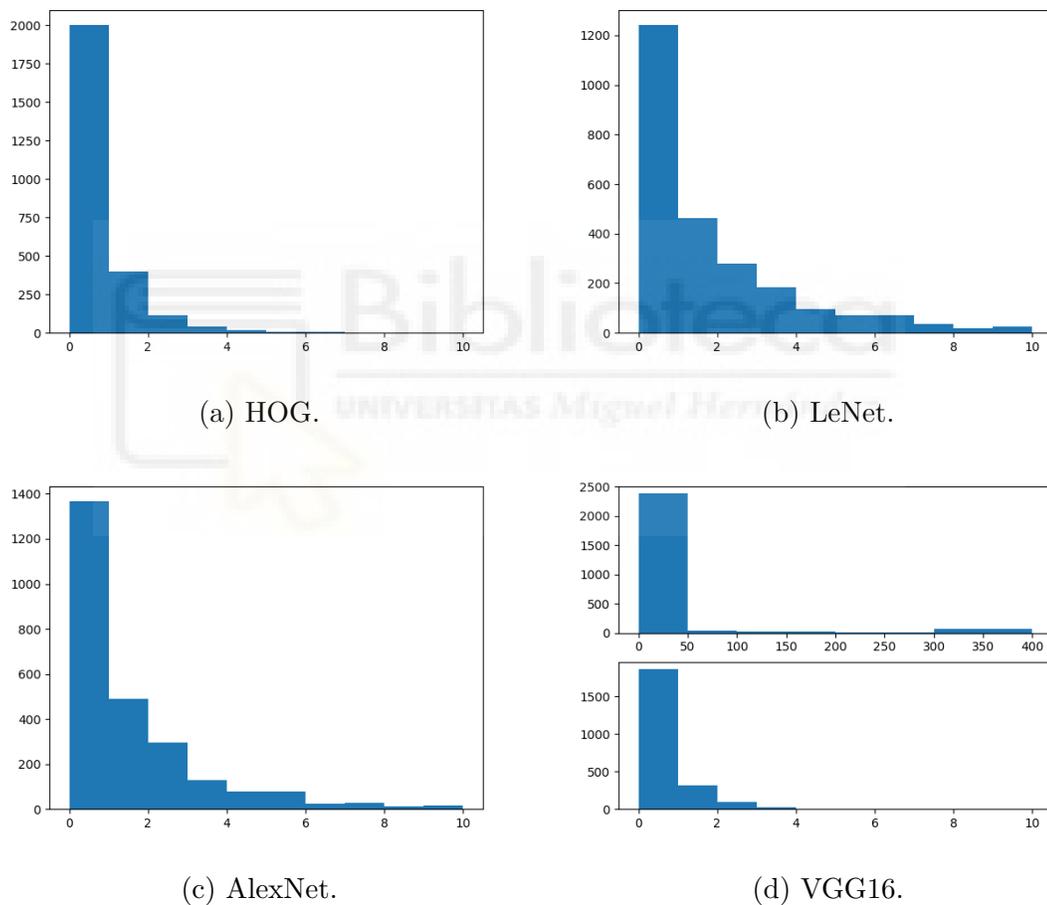


Figura 4-3: Histogramas de proximidad de imagen seleccionada.

4.2.2. Localización jerárquica

En este experimento, se compara el descriptor de cada imagen de testeo con los descriptores representativos para encontrar el cluster correspondiente. Luego se compara el descriptor test con todos los descriptores de las imágenes contenidos en el cluster seleccionado.

Para ello se realiza el clustering espectral, elaborando previamente el modelo del entorno

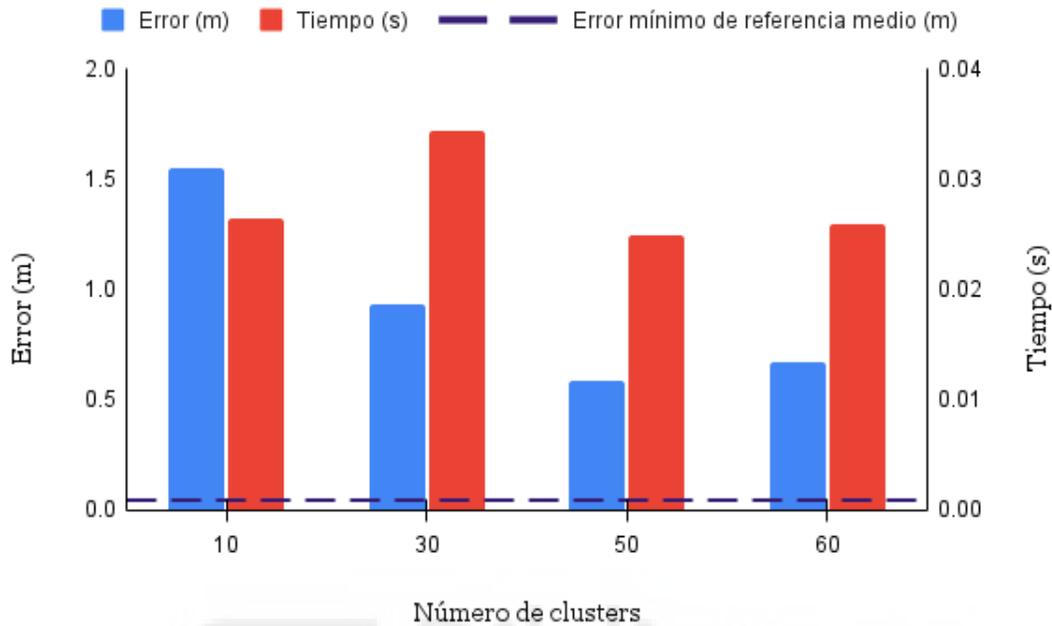


Figura 4-4: Resultados del método jerárquico.

con el descriptor *gist* implementado en MATLAB para cada imagen del conjunto de mapeo. Se selecciona un número de clusters y automáticamente se le asignan etiquetas a cada imagen.

Después, se construye el modelo jerárquico con el descriptor HOG, asignando a cada descriptor su etiqueta correspondiente, y obteniendo los descriptores representativos de cada cluster como la media de todos los descriptores con la misma etiqueta. Así, durante el experimento, se obtiene el descriptor HOG de cada imagen de test y se aplica el algoritmo.

En la figura 4-4 se puede ver los resultados obtenidos con distintos números de clusters. Para 50 clusters se obtiene el mejor resultado de tiempo y error de localización, generando el mapa que se muestra en la figura 4-5.

4.2.3. Localización probabilística basada en filtro bayesiano

En este experimento se prueba el algoritmo basado en el filtro de Bayes, con un modelo movimiento que se usa para filtrar posteriormente la información obtenida por modelo el observación.

La forma de simular el movimiento entre poses consecutivas del robot consiste en calcular la diferencia entre dichas posiciones, de manera que el comando de control queda como $\vec{u} = [x_f - x_o, y_f - y_o]$ al que se le suma una cierta cantidad de ruido gaussiano para obtener una fdp de las posiciones del modelo del entorno con valores inversamente proporcionales a la distancia a la posición de destino tras el movimiento. A continuación, se aplica el modelo de observación, comparando con todos los descriptores del entorno.

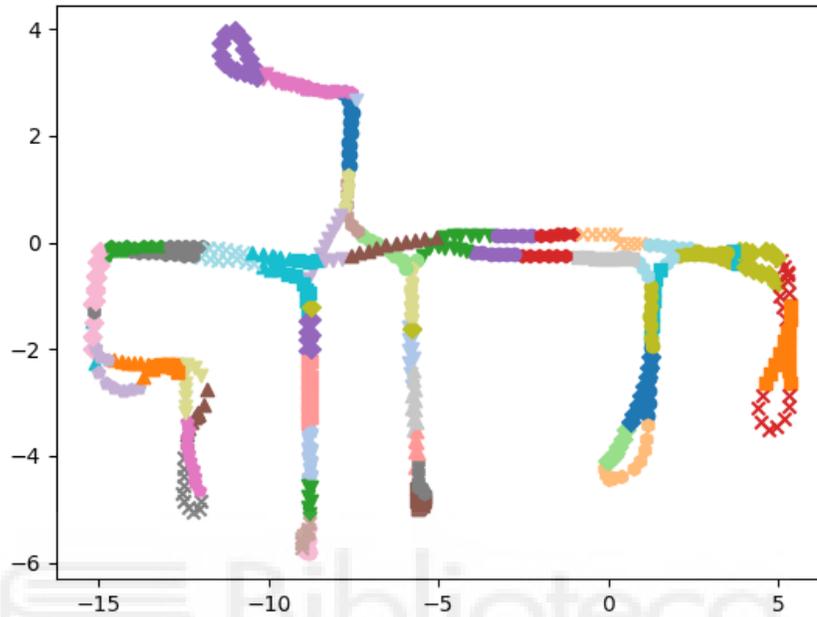


Figura 4-5: Mapa de Friburgo separado en 50 clusters.

Para obtener la estimación de la posición final en cada iteración, se toman las posiciones con las probabilidades más altas, y se hace una media ponderada. Se estudiará cuántas posiciones es bueno considerar a la hora de calcular esta media. Para ello, También se empleará un mapa de tipo rejilla, además del mapa de trayectoria que se ha usado hasta ahora, a fin de determinar si en distintas configuraciones es mejor considerar más o menos posiciones. Es decir, cabría esperar que en el mapa rejilla, al tener posiciones de referencia de mapeo en todas direcciones, fuera más ventajoso ponderar más posiciones para la estimación final.

Mapa rejilla

El mapa de rejilla a emplear se elabora con la base de datos Quorum V [50], que cuenta con conjuntos de imágenes de diferentes estancias del edificio homónimo de la universidad Miguel Hernández de Elche. Para este experimento, se usará solamente el conjunto de imágenes del salón de eventos, que incluye una serie de imágenes equidistantes entre ellas, formando una rejilla cuadrada. Además, se tomaron también algunas imágenes en posiciones fuera de la rejilla, de modo que tomando estas imágenes, junto con algunas de la propia rejilla, se puede simular el seguimiento de una trayectoria en este entorno, tal y como se muestra en la figura 4-6. Están disponibles las imágenes tanto en formato omnidireccional como panorámico, y se usará directamente el segundo conjunto para hacer el proceso más similar al resto con las imágenes de Friburgo.

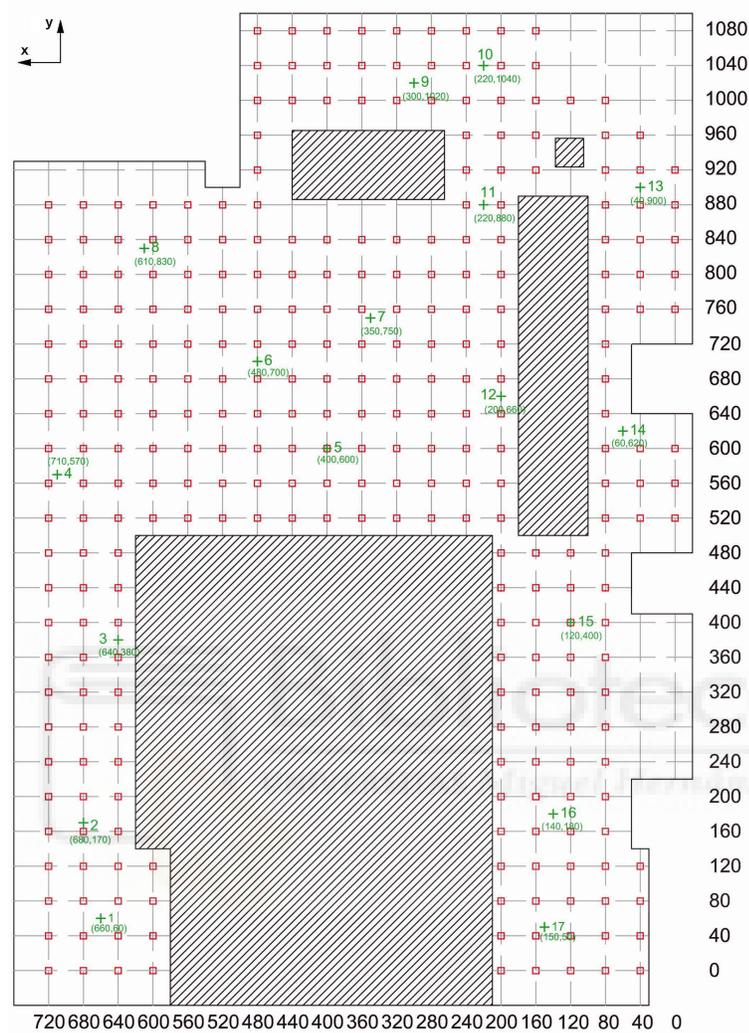


Figura 4-6: Mapa rejilla. Fuente [51]

En las figuras 4-7 y 4-8 se muestran los resultados de haber realizado el experimento en el mapa de trayectoria y de rejilla respectivamente, ambos con el descriptor HOG. Las medidas de error obtenidas no son comparables entre ellas debido al diferente espaciado y disposición de las imágenes de mapeo, pero sí se puede analizar la diferente medida de error obtenida al considerar mayor o menor cantidad de imágenes la estimación final de la posición, resultando en ambos casos que se obtiene el mejor resultado al tomar las coordenadas ponderadas de las dos posiciones más probables del modelo.

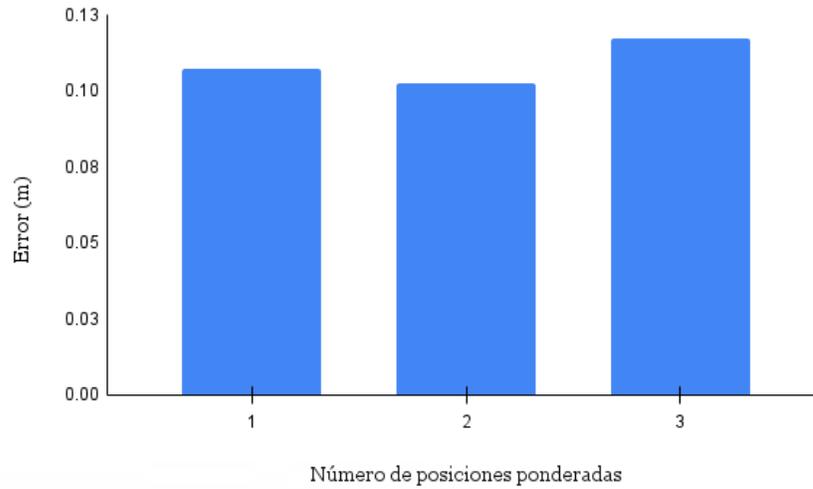


Figura 4-7: Resultados del método probabilístico. Mapa de trayectoria.

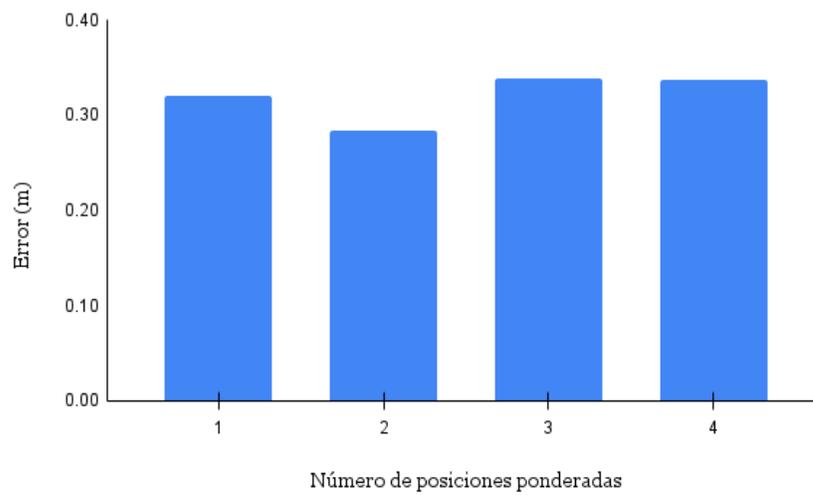


Figura 4-8: Resultados del método probabilístico. Mapa de rejilla.

4.2.4. Localización jerárquica probabilística

Este experimento aplica el modelo de movimiento igual que el anterior, pero en el modelo de observación, en lugar de comparar con todas las imágenes del modelo, se hace una localización jerárquica, donde la localización gruesa se lleva a cabo a través de la clasificación realizada por la CNN, que será la versión reentrenada de AlexNet. Con esta metodología se compara el funcionamiento empleando el descriptor HOG, así como el obtenido de una capa intermedia de la propia CNN clasificadora. En este caso solamente se realiza el experimento sobre el mapa de trayectoria de Friburgo, obteniendo los resultados expuestos en la figura 4-9. Los resultados obtenidos son muy similares, pero igual que en el caso anterior, ponderando las dos posiciones con mayor probabilidad se minimiza el error.

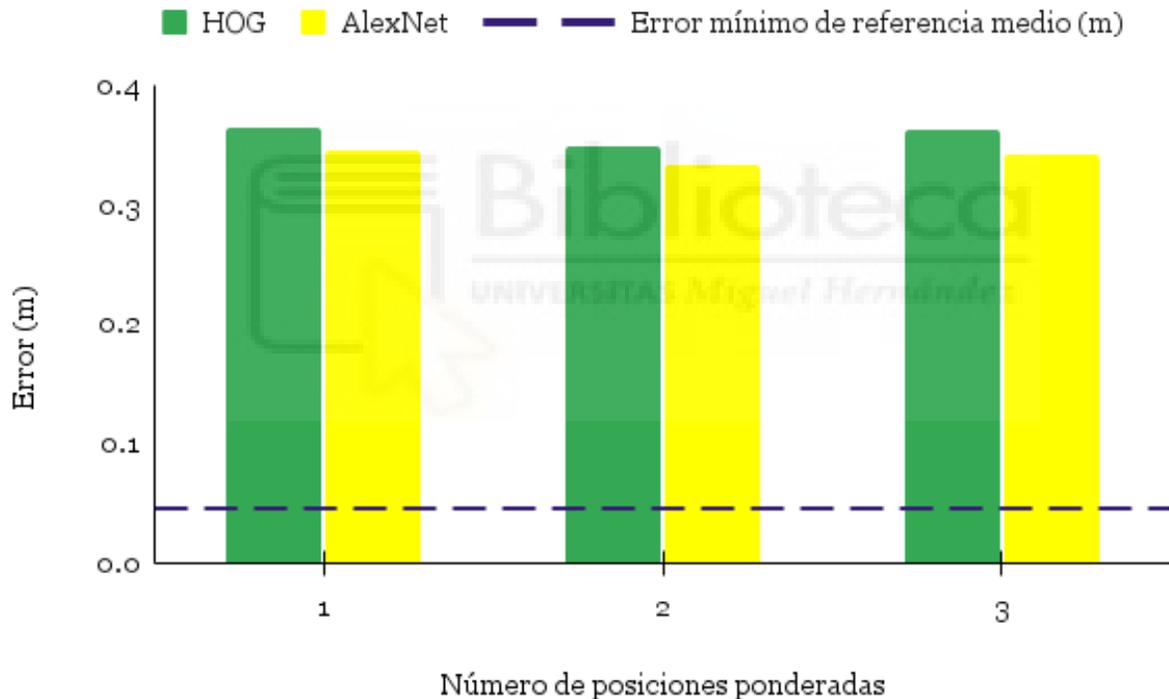


Figura 4-9: Resultados del método jerárquico probabilístico.

4.2.5. Localización probabilística con y sin métrica

En este último experimento, se comparan los dos últimos algoritmos, el que tiene en cuenta información métrica de confianza, y el que no considera ningún tipo de información métrica. Todos los resultados se han obtenido empleando el descriptor HOG, que es el que ha dado mejores resultados en general.

Para el método sin información métrica, se ha cogido una ventana de búsqueda de ± 3 posiciones alrededor, y como las imágenes están espaciadas de media unos 20 cm, sería equi-

valente a tomar una amplitud de búsqueda en un espacio de 120cm; este método no incluye ningún tipo de perturbación en el modelo de movimiento, debido a cómo está planteado. Para el método que tiene en cuenta la información métrica, se prueba con varios pares de umbrales en cm: 20/60, 10/30 y 60/100 como umbrales 1, 2 y 3, respectivamente; así como distintos niveles de ruido en el modelo de movimiento para cada umbral, siendo en cm una desviación típica de 8, 20, 40 y 100. Se muestran los resultados obtenidos en la figura 4-10, con una vista detallada en la figura 4-11.

Como se puede apreciar, el método que no incluye información métrica tiene un error constante pues no le afecta el ruido del modelo de movimiento; mientras que el método con información métrica es sensible al ruido y su medida de error aumenta junto con el nivel del ruido.

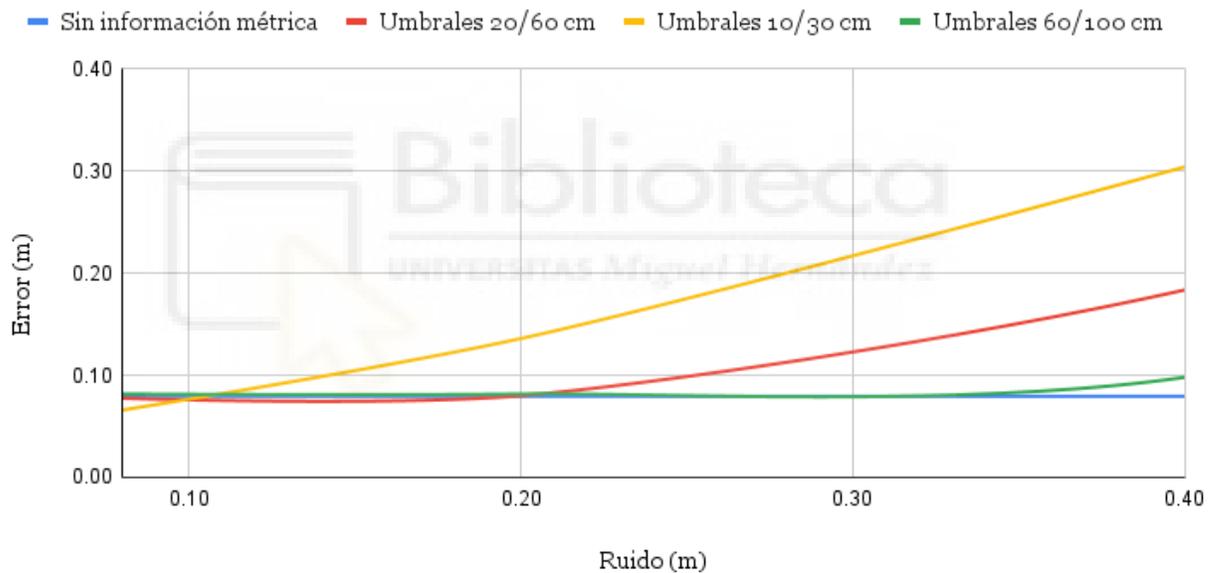


Figura 4-10: Comparativa de resultados de los métodos con y sin información métrica.

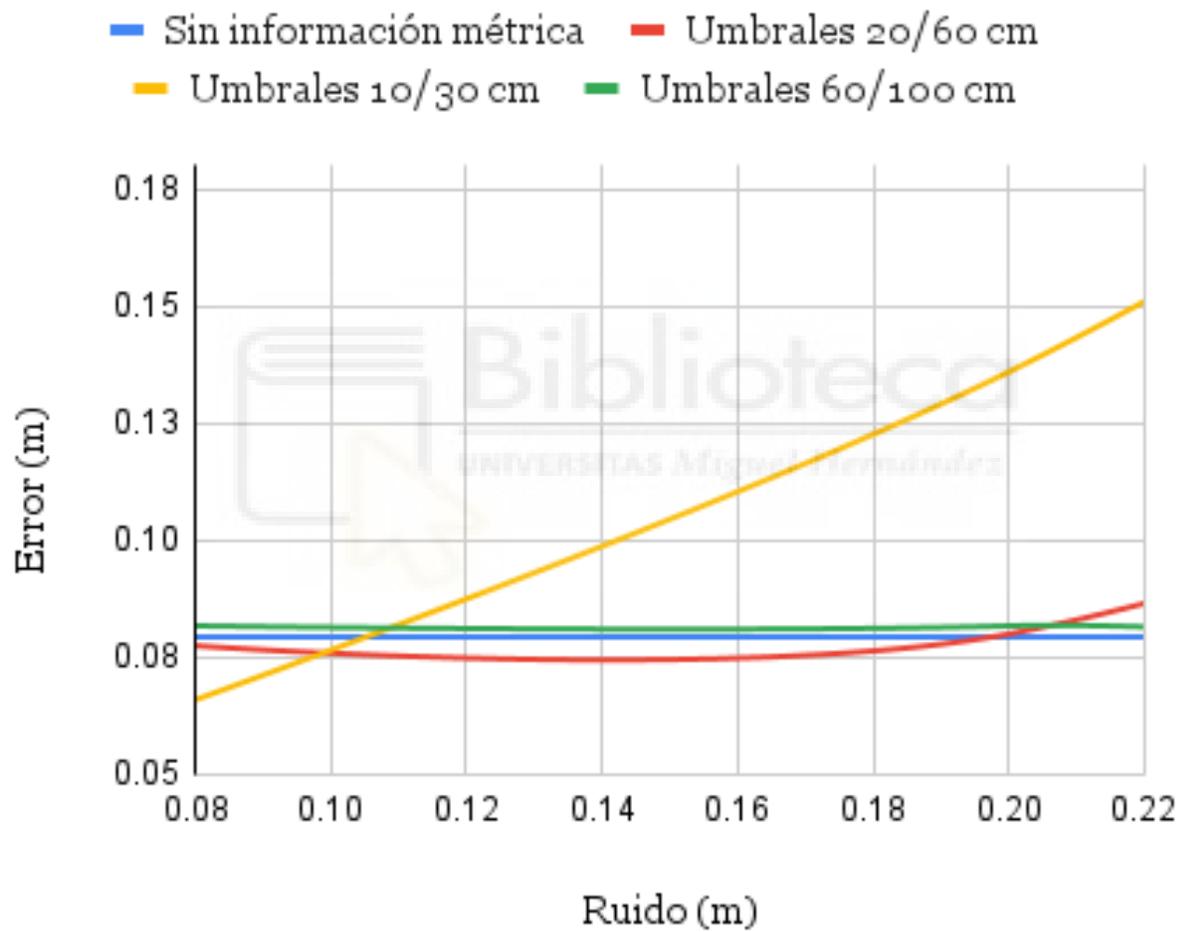


Figura 4-11: Comparativa de resultados de los métodos con y sin información métrica. Detalle.

5 Conclusiones y trabajos futuros

En el presente Trabajo de Fin de Máster se han estudiado los problemas en la tarea de localización empleando diversos enfoques y técnicas para el desarrollo de algoritmos que permitan resolver estos problemas. Tras analizar los resultados obtenidos en los experimentos realizados, se sacan una serie de conclusiones y se plantean posibles líneas de trabajo para expandir este trabajo en el futuro.

5.1. Conclusiones

- De entre los descriptores utilizados en los experimentos, el descriptor HOG ha sido el que ha dado mejores resultados tanto en error de localización como en tiempo de cómputo. Los descriptores extraídos de capas intermedias de las CNN no han dado resultados comparables, pero puede ser interesante emplearlos en caso de que se use la misma CNN también para la tarea de clasificación. También se observa una cierta correlación entre la tasa de aciertos de la CNN en la tarea de localización y el error de localización cometido por el descriptor generado.
- Respecto a las CNN estudiadas en este trabajo, AlexNet da mejores resultados al ser reentrenada para estas tareas, tanto clasificación como generación de descriptores; quizás debido a una arquitectura más compleja que le permite aprender mejor sobre las imágenes a clasificar. No obstante, aún cabe la posibilidad de explorar otras arquitecturas o incluso diseñar una específica para esta tarea.
- Los métodos jerárquicos han demostrado ser una forma de reducir el tiempo de cómputo en la tarea de localización. Sin embargo, los propuestos en este trabajo aún cometen un error de localización considerable, debido a una incorrecta asignación del cluster en la etapa de localización gruesa. Si se consigue reducir el error lo suficiente, los métodos jerárquicos se convierten en una solución de compromiso razonable cuando una aplicación precisa reducir el tiempo de cómputo.
- Los métodos probabilísticos, por otro lado, sí que han dado buenos resultados y permiten dar una estimación más realista de la posición del robot, que no tiene que estar limitada a las posiciones discretas del modelo. Se observa que se obtiene el mejor resultado ponderando las coordenadas de las dos posiciones más probables obtenidas por el algoritmo. Para lograr esto, se añade información adicional al algoritmo mediante

la inclusión del modelo de movimiento, que podrá ser más o menos precisa según las prestaciones de los sensores que se utilicen para obtener esta información, adicionales a la cámara omnidireccional que es el foco de los experimentos.

- En general, de todos los algoritmos estudiados, el del vecino más cercano es simple de implementar y da buenos resultados, necesitando solamente la cámara para funcionar. Sin embargo, merece la pena implementar el método probabilístico sin información métrica, que obtiene mejores resultados y tampoco requiere sensores adicionales, aunque sí que requiere tener la seguridad de que el robot no se va a desplazar demasiado entre instantes de muestreo. Por otro lado, si se dispone de sensores especialmente precisos como LiDAR o GPS para el modelo de movimiento, se puede obtener el mejor resultado incluyendo esta información en el modelo de movimiento, pero hay que atender a las capacidades de estos sensores, pues este método es óptimo con niveles de ruido bajos, pero el error crece rápidamente con el ruido.

5.2. Trabajos futuros

Del presente Trabajo de Fin de Máster se proponen posibles líneas futuras de investigación que guardan relación con el mismo:

- En primer lugar, sería interesante solucionar el problema de la localización gruesa para mejorar el rendimiento de los métodos jerárquicos. Para ello, sería interesante explorar la posibilidad de hacer diferentes agrupamientos, quizás con distinto número de clusters, o con un criterio diferente al descriptor gist utilizado en este trabajo.
- Se podrían emplear técnicas de Inteligencia Artificial para solucionar este problema mencionado, esto es realizar un clustering incremental, en el que una red neuronal es entrenada para clasificar las imágenes, sin haber predefinido las clases, sino que durante el propio entrenamiento la red separa todo el conjunto de imágenes en grupos.

También se puede seguir con el mismo enfoque empleado, pero probar a entrenar otras CNN, o diseñar una arquitectura nueva para este problema, o incluso emplear otro tipo de redes como los *transformers*

Respecto al entrenamiento de las redes, también sería interesante entrenar con conjuntos de imágenes con variaciones como cambios de iluminación, u otras bases de datos de otros entornos para comprobar si estos métodos son generalizables a otros entornos y otras condiciones de iluminación. También probar en entornos de exteriores.

- Por último, el siguiente paso sería probar estos algoritmos en directo, con un robot en movimiento. Para ello, sería preciso la elaboración de nuevas bases de datos de imágenes del entorno donde se vaya a probar. Esto permitiría fácilmente tomar imágenes en una

trayectoria desfasada a la del modelo, o implementar un modelo de movimiento que se ajuste mejor a las características reales del robot.

Además, en estos nuevos experimentos, se podría explorar la posibilidad de incluir información de una mayor variedad de sensores, e integrarla en los algoritmos existentes, o en nuevos algoritmos, basados en filtro de Kalman por ejemplo, para refinar el proceso de localización.



Bibliografía

- [1] Antoni Grau et al. “Industrial robotics in factory automation: From the early stage to the Internet of Things”. En: *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. 2017, págs. 6159-6164. DOI: 10.1109/IECON.2017.8217070.
- [2] Fahad Mehfuz. “Recent Implementations of Autonomous Robotics for Space Exploration”. En: *2018 International Conference on Sustainable Energy, Electronics, and Computing Systems (SEEMS)*. 2018, págs. 1-6. DOI: 10.1109/SEEMS.2018.8687338.
- [3] Kittit Thamrongaphichartkul et al. “A Framework of IoT Platform for Autonomous Mobile Robot in Hospital Logistics Applications”. En: *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*. 2020, págs. 1-6. DOI: 10.1109/iSAI-NLP51646.2020.9376823.
- [4] R. A. Saeed et al. “Metrology-aware Path Planning for Agricultural Mobile Robots in Dynamic Environments”. En: *2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*. 2021, págs. 448-453. DOI: 10.1109/MetroAgriFor52389.2021.9628737.
- [5] Alberto Luvisutto et al. “Robotic Swarm for Marine and Submarine Missions: Challenges and Perspectives”. En: *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*. 2022, págs. 1-8. DOI: 10.1109/AUV53081.2022.9965934.
- [6] G. Giralt, R. Sobek y R. Chatila. “A multi-level planning and navigation system for a mobile robot: a first approach to HILARE”. En: *Proceedings of the 6th International Joint Conference on Artificial Intelligence (1979)*, págs. 335-337.
- [7] Zeyuan Sun et al. “BIT-DMR: A Humanoid Dual-Arm Mobile Robot for Complex Rescue Operations”. En: *IEEE Robotics and Automation Letters* 7.2 (2022), págs. 802-809. DOI: 10.1109/LRA.2021.3131379.
- [8] Jingdao Chen y Yong K. Cho. “Detection of Damaged Infrastructure on Disaster Sites using Mobile Robots”. En: *2019 16th International Conference on Ubiquitous Robots (UR)*. 2019, págs. 648-653. DOI: 10.1109/URAI.2019.8768770.
- [9] Cosmin Marcu y Levente Tamas. “Navigation of Outdoor Mobile Robots with Extended Grid Algorithms”. En: *2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*. 2020, págs. 1-6. DOI: 10.1109/AQTR49680.2020.9129936.

- [10] Sukkpranhachai Gatesichapakorn, Jun Takamatsu y Miti Ruchanurucks. “ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera”. En: *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*. 2019, págs. 151-154. DOI: 10.1109/ICA-SYMP.2019.8645984.
- [11] Kanin Piemngam, Itthisek Nilkhamhang y Pished Bunnun. “Development of Autonomous Mobile Robot Platform with Mecanum Wheels”. En: *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*. 2019, págs. 90-93. DOI: 10.1109/ICA-SYMP.2019.8646085.
- [12] Chandrakant Supare y S D Adlinge. “Control of Mobile Robot Using Visual Feedback and Wireless Communication”. En: *2018 3rd International Conference for Convergence in Technology (I2CT)*. 2018, págs. 1-5. DOI: 10.1109/I2CT.2018.8529367.
- [13] Satria Rachmad Santosa, Bima Sena Bayu Dewantara e Iwan Kurnianto Wibowo. “Ball Detection Under Variety Lighting using Omnidirectional Camera”. En: *2019 International Electronics Symposium (IES)*. 2019, págs. 235-240. DOI: 10.1109/ELECSYM.2019.8901617.
- [14] Chinthaka Premachandra y Masaya Tamaki. “A Hybrid Camera System for High-Resolutionization of Target Objects in Omnidirectional Images”. En: *IEEE Sensors Journal* 21.9 (2021), págs. 10752-10760. DOI: 10.1109/JSEN.2021.3059102.
- [15] Tomohiro Shimizu y Kazuyuki Kobayashi. “Development of a Person-Searching Algorithm Using an Omnidirectional Camera and LiDAR for the Tsukuba Challenge”. En: *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 2018, págs. 810-815. DOI: 10.23919/SICE.2018.8492678.
- [16] Yuya Kato, Junichi Hara e Hiroshi Watanabe. “Creation of 3D Environmental Map using Omnidirectional Camera Images”. En: *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*. 2021, págs. 450-453. DOI: 10.1109/GCCE53005.2021.9621940.
- [17] Alberto Torres Angonese y Paulo Fernando Ferreira Rosa. “Integration of People Detection and Simultaneous Localization and Mapping Systems for an Autonomous Robotic Platform”. En: *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*. 2016, págs. 251-256. DOI: 10.1109/LARS-SBR.2016.49.
- [18] A. C. Murillo, J. J. Guerrero y C. Sagues. “SURF features for efficient robot localization with omnidirectional images”. En: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, págs. 3901-3907. DOI: 10.1109/ROBOT.2007.364077.
- [19] Lin Ma et al. “A Fast C-GIST Based Image Retrieval Method for Vision-Based Indoor Localization”. En: *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. 2017, págs. 1-5. DOI: 10.1109/VTCSpring.2017.8108338.

- [20] Saprem Shah et al. “Unsupervised GIST based Clustering for Object Localization”. En: *2019 National Conference on Communications (NCC)*. 2019, págs. 1-6. DOI: 10.1109/NCC.2019.8732251.
- [21] Jingwen Zhao et al. “Real-time human detection with depth camera via a physical radius-depth detector and a CNN descriptor”. En: *2017 IEEE International Conference on Multimedia and Expo (ICME)*. 2017, págs. 1536-1541. DOI: 10.1109/ICME.2017.8019323.
- [22] Frazer K. Noble. “A mobile robot platform for supervised machine learning applications”. En: *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. 2017, págs. 1-6. DOI: 10.1109/M2VIP.2017.8211472.
- [23] T. Tongloy et al. “Asynchronous deep reinforcement learning for the mobile robot navigation with supervised auxiliary tasks”. En: *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*. 2017, págs. 68-72. DOI: 10.1109/ICRAE.2017.8291355.
- [24] Anastasia Gaydashenko, Daniel Kudenko y Aleksei Shpilman. “A Comparative Evaluation of Machine Learning Methods for Robot Navigation Through Human Crowds”. En: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, págs. 553-557. DOI: 10.1109/ICMLA.2018.00089.
- [25] Chunling Liu, Jun Xu y Kaiwen Guo. “Path Planning for Mobile Robot Based on Deep Reinforcement Learning and Fuzzy Control”. En: *2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*. 2022, págs. 533-537. DOI: 10.1109/ICICML57342.2022.10009792.
- [26] O. J. Céspedes. “Localización de un robot móvil utilizando información visual y redes neuronales convolucionales”. En: *Trabajo de Fin de Grado, Ingeniería de Sistemas y Automática, Universidad Miguel Hernández de Elche* (2020).
- [27] Francisco Amorós et al. “Holistic Descriptors of Omnidirectional Color Images and Their Performance in Estimation of Position and Orientation”. En: *IEEE Access* 8 (2020), págs. 81822-81848. DOI: 10.1109/ACCESS.2020.2990996.
- [28] V. Román et al. “Evaluating the Robustness of New Holistic Description Methods in Position Estimation of Mobile Robots”. En: *International Conference on Informatics in Control, Automation and Robotics* (2022), págs. 226-246. DOI: 10.1007/978-3-030-92442-3_12.
- [29] Sergio Cebollada et al. “A Localization Approach Based on Omnidirectional Vision and Deep Learning”. En: *Informatics in Control, Automation and Robotics*. Ed. por Oleg Gusikhin, Kurosh Madani y Janan Zaytoon. Cham: Springer International Publishing, 2022, págs. 226-246. ISBN: 978-3-030-92442-3.

- [30] S. Cebollada et al. “Hierarchical Localization in Topological Models Under Varying Illumination Using Holistic Visual Descriptors.” En: *IEEE Access* 7 (2019).
- [31] David Valiente et al. “Dynamic Catadioptric Sensory Data Fusion for Visual Localization in Mobile Robotics”. En: *Proceedings* 15.1 (2019). ISSN: 2504-3900. DOI: 10.3390/proceedings2019015002. URL: <https://www.mdpi.com/2504-3900/15/1/2>.
- [32] Fengbing Luo, Bianjing Du y Zhen Fan. “Mobile robot localization based on particle filter”. En: *Proceeding of the 11th World Congress on Intelligent Control and Automation*. 2014, págs. 3089-3093. DOI: 10.1109/WCICA.2014.7053222.
- [33] M. Xu, N. Snderhauf y M. Milford. “Probabilistic Visual Place Recognition for Hierarchical Localization”. En: *IEEE Robotics and Automation Letters* 6(2) (2021), págs. 311-318.
- [34] Zhongwei Hua y Dongdong He. “Multi-Sensor Fusion Localization and Mapping of Indoor Mobile Robot”. En: *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*. 2022, págs. 7-11. DOI: 10.1109/AEMCSE55572.2022.00008.
- [35] Zhuang Dai et al. “CNN Descriptor Improvement Based on L2-Normalization and Feature Pooling for Patch Classification”. En: *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2018, págs. 144-149. DOI: 10.1109/ROBIO.2018.8665330.
- [36] Yu Lin. “Wafer Pattern Counting, Detection and Classification Based on Encoder-Decoder CNN Structure”. En: *2022 IEEE International Conference on Electro Information Technology (eIT)*. 2022, págs. 109-113. DOI: 10.1109/eIT53891.2022.9813870.
- [37] O. Hmidani y E. M. Ismaili Alaoui. “A comprehensive survey of the R-CNN family for object detection”. En: *2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet)*. 2022, págs. 1-6. DOI: 10.1109/CommNet56067.2022.9993862.
- [38] J. Cabrera et al. “Training, Optimization and Validation of a CNN for Room Retrieval and Description of Omnidirectional Images”. En: *SN Computer Science* 3 (2022).
- [39] Jun Ding et al. “Convolutional Neural Network With Data Augmentation for SAR Target Recognition”. En: *IEEE Geoscience and Remote Sensing Letters* 13.3 (2016), págs. 364-368. DOI: 10.1109/LGRS.2015.2513754.
- [40] Yiming Wan et al. “Boosting Image-Based Localization Via Randomly Geometric Data Augmentation”. En: *2020 IEEE International Conference on Image Processing (ICIP)*. 2020, págs. 688-692. DOI: 10.1109/ICIP40778.2020.9190809.

- [41] Lili Zhang, Xiulei Liu y Qiang Tong. “ECopy-Paste: An Effective Data Augmentation Method for Object Detection of UAV”. En: *2022 International Conference on Big Data, Information and Computer Network (BDICN)*. 2022, págs. 469-474. DOI: 10.1109/BDICN55575.2022.00092.
- [42] Shohei Chiba e Hisayuki Sasaoka. “Basic Study for Transfer Learning for Autonomous Driving in Car Race of Model Car”. En: *2021 6th International Conference on Business and Industrial Research (ICBIR)*. 2021, págs. 138-141. DOI: 10.1109/ICBIR52339.2021.9465856.
- [43] Daniel Tan Wei Xun, Yoke Lin Lim y Sutthiphong Srigrarom. “Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2”. En: *2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*. 2021, págs. 1-6. DOI: 10.1109/ICA-SYMP50206.2021.9358449.
- [44] Andrzej Pronobis y Barbara Caputo. “COLD: COsy Localization Database”. En: *The International Journal of Robotics Research (IJRR)* 28.5 (mayo de 2009), págs. 588-594. DOI: 10.1177/0278364909103912. URL: <http://www.pronobis.pro/publications/pronobis2009ijrr>.
- [45] <https://www.cas.kth.se/COLD/cold-freiburg.html>. Accessed: 2022-06-13.
- [46] <https://scikit-learn.org/stable/modules/clustering.html>. Accessed: 2022-06-13.
- [47] Y. Lecun et al. “Gradient-based learning applied to document recognition”. En: *Proceedings of the IEEE* 86.11 (1998), págs. 2278-2324. DOI: 10.1109/5.726791.
- [48] Alex Krizhevsky, Ilya Sutskever y Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. En: *Commun. ACM* 60.6 (mayo de 2017), págs. 84-90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: <https://doi.org/10.1145/3065386>.
- [49] <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>. Accessed: 2022-06-13.
- [50] Luis Payá et al. “Performance of Global-Appearance Descriptors in Map Building and Localization Using Omnidirectional Vision”. En: *Sensors* 14.2 (2014), págs. 3033-3064. ISSN: 1424-8220. DOI: 10.3390/s140203033. URL: <https://www.mdpi.com/1424-8220/14/2/3033>.
- [51] <https://arvc.umh.es/db/images/quorumv/>. Accessed: 2022-06-13.