Programa de Doctorado en Tecnologías Industriales y de Telecomunicación

Creation and maintenance of visual incremental maps and hierarchical localization.

Vicente Román Erades

Director de la tesis

Dr. D. Óscar Reinoso García

Codirector de la tesis

Dr. D. Luis Payá Castelló

Universidad Miguel Hernández de Elche

- 2022 –

La presente Tesis Doctoral, titulada “Creation and maintenance of visual incremental maps and hierarchical localization.”, se presenta bajo la modalidad de **tesis por compendio** de las siguientes **publicaciones**:

- The Role of Global Appearance of Omnidirectional Images in Relative Distance and Orientation Retrieval
  V. Román, L. Payá, A. Peidró, M. Ballesta, O. Reinoso
  Sensors. 21(10), 3327 (Mayo 2021)
  ISSN:1424-8220. Ed. MDPI
  JCR-SCI Impact Factor(2020, último dato disponible en el momento de la aceptación del artículo): 3.576, Quartile Q1
  Web: https://doi.org/10.3390/s21103327
  DOI: 10.3390/s21103327

- Creating Incremental Models of Indoor Environments through Omnidirectional Imaging.
  V. Román, L. Payá, S. Cebollada, O. Reinoso
  Applied Sciences. Vol 10(18),6480 (Septiembre 2020)
  ISSN:2076-3417. Ed. MDPI
  JCR-SCI Impact Factor(2020, último dato disponible en el momento de la aceptación del artículo): 2.679, Quartile Q2
  Web: https://doi.org/10.3390/app10186480
  DOI: 10.3390/app10186480

El Dr. D. Óscar Reinoso García, director, y el Dr. D. Luis Payá Castelló, codirector de la tesis doctoral titulada **"Creation and maintenance of visual incremental maps and hierarchical localization."**

INFORMAN:

Que D. Vicente Román Erades ha realizado bajo nuestra supervisión el trabajo titulado **"Creation and maintenance of visual incremental maps and hierarchical localization"** conforme a los términos y condiciones definidos en su Plan de Investigación y de acuerdo al Código de Buenas Prácticas de la Universidad Miguel Hernández de Elche, cumpliendo los objetivos previstos de forma satisfactoria para su defensa pública como tesis doctoral.

Lo que firmamos para los efectos oportunos, en Elche a de de 2022.

Director de la tesis
Dr. D. Óscar Reinoso García

Codirector de la tesis
Dr. D. Luis Payá Castelló

El Dr. D. Óscar Reinoso García, Coordinador del Programa de Doctorado en Industriales y de Telecomunicación de la Universidad Miguel Hernández de Elche.

INFORMA:

Que D. Vicente Román Erades ha realizado bajo la supervisión de nuestro Programa de Doctorado el trabajo titulado **"Creation and maintenance of visual incremental maps and hierarchical localization."** conforme a los términos y condiciones definidos en su Plan de Investigación y de acuerdo al Código de Buenas Prácticas de la Universidad Miguel Hernández de Elche, cumpliendo los objetivos previstos de forma satisfactoria para su defensa pública como tesis doctoral.

Lo que firmo para los efectos oportunos, en Elche a de de 2022.

Prof. Dr. D. Óscar Reinoso García

Coordinador del Programa de Doctorado en Industriales y de Telecomunicación de la Universidad Miguel Hernández de Elche.

Abstract

Over the last few years, the presence of the mobile robotics has considerably increased in a wide variety of environments. It is common to find robots that carry out repetitive and specific applications and also, they can be used for working at dangerous environments and to perform precise tasks. These robots can be found in a variety of social environments, such as industry, household, educational and health scenarios. For that reason, they need a specific and continuous research and improvement work. Specifically, autonomous mobile robots require a very precise technology to perform tasks without human assistance.

To perform tasks autonomously, the robots must be able to navigate in an unknown environment. For that reason, the autonomous mobile robots must be able to address the mapping and localization tasks: they must create a model of the environment and estimate their position and orientation.

This PhD thesis proposes and analyses different methods to carry out the map creation and the localization tasks in indoor environments. To address these tasks only visual information is used, specifically, omnidirectional images, with a 360º field of view. Throughout the chapters of this document solutions for autonomous navigation tasks are proposed, they are solved using transformations in the images captured by a vision system mounted on the robot.

Firstly, the thesis focuses on the study of the global appearance descriptors in the localization task. The global appearance descriptors are algorithms that transform an image globally, into a unique vector. In these works, a deep comparative study is performed. In the experiments different global appearance descriptors are used along with omnidirectional images and the results are compared. The main goal is to obtain an optimized algorithm to estimate the robot position and orientation in real indoor environments. The experiments take place with real conditions, so some visual changes in the scenes can occur, such as camera defects, furniture or people movements and changes in the lighting conditions. The computational cost is also studied; the idea is that the robot has to localize the robot in an accurate mode, but also, it has to be fast enought.

Additionally, a second application, whose goal is to carry out an incremental mapping in indoor environments, is presented. This application uses the best global appearance descriptors used in the localization task, but this time they are constructed with the purpose of solving the mapping problem using an incremental clustering technique. The application clusters a batch of images that are visually similar; every group of images or cluster is expected to identify a zone of the environment. The shape and size of the cluster can vary while the robot is visiting the different rooms. Nowadays. different algorithms can be used to obtain the clusters, but all these solutions usually work properly when they work ‘off-line’, starting from the whole set of data to cluster. The main idea of this study is to obtain the map incrementally while the robot explores the new environment. Carrying out the mapping incrementally while the robot is still visiting the area is

very interesting since having the map separated into nodes with relationships of similitude between them can be used subsequently for the hierarchical localization tasks, and also, to recognize environments already visited in the model.

Finally, this PhD thesis includes an analysis of deep learning techniques for localization tasks. Particularly, siamese networks have been studied. Siamese networks are based on classic convolutional networks, but they permit evaluating two images simultaneously. These networks output a similarity value between the input images, and that information can be used for the localization tasks. Throughout this work the technique is presented, the possible architectures are analysed and the results after the experiments are shown and compared. Using the siamese networks, the localization in real operation conditions and environments is solved, focusing on improving the performance against illumination changes on the scene. During the experiments the room retrieval problem, the hierarchical localization and the absolute localization have been solved.

Resumen

Durante los últimos años, la presencia de la robótica móvil ha aumentado substancialmente en una gran variedad de entornos y escenarios. Es habitual encontrar el uso de robots para llevar a cabo aplicaciones repetitivas y específicas, así como tareas en entornos peligrosos o con resultados que deben ser muy precisos. Dichos robots se pueden encontrar tanto en ámbitos industriales como en familiares, educativos y de salud; por ello, requieren un trabajo específico y continuo de investigación y mejora. En concreto, los robots móviles autónomos requieren de una tecnología precisa para desarrollar tareas sin ayuda del ser humano.

Para realizar tareas de manera autónoma, los robots deben ser capaces de navegar por un entorno 'a priori' desconocido. Por tanto, los robots móviles autónomos deben ser capaces de realizar la tarea de creación de mapas, creando un modelo del entorno y la tarea de localización, esto es estimar su posición y orientación.

La presente tesis plantea un diseño y análisis de diferentes métodos para realizar las tareas de creación de mapas y localización en entornos de interior. Para estas tareas se emplea únicamente información visual, en concreto, imágenes omnidireccionales, con un campo de visión de 360º. En los capítulos de este trabajo se plantean soluciones a las tareas de navegación autónoma del robot mediante transformaciones en las imágenes que este es capaz de captar.

En cuanto a los trabajos realizados, en primer lugar, se presenta un estudio de descriptores de apariencia global en tareas de localización. Los descriptores de apariencia global son transformaciones capaces de obtener un único vector que describa globalmente una imagen. En este trabajo se realiza un estudio exhaustivo de diferentes métodos de apariencia global adaptando su uso a imágenes omnidireccionales. Se trata de obtener un algoritmo optimizado para estimar la posición y orientación del robot en entornos reales de oficina, donde puede surgir cambios visuales en el entorno como movimientos de cámara, de mobiliario o de iluminación en la escena. También se evalúa el tiempo empleado para realizar esta estimación, ya que el trabajo de un robot debe ser preciso, pero también factible en cuanto a tiempos de computación.

Además, se presenta una segunda aplicación donde el estudio se centra en la creación de mapas de entornos de interior de manera incremental. Esta aplicación hace uso de los descriptores de apariencia global estudiados para la tarea de localización, pero en este caso se utilizan para la construcción de mapas utilizando la técnica de 'clustering' incremental. En esta aplicación, conjuntos de imágenes visualmente similares se agrupan en un único grupo. La forma y cantidad de grupos es variable conforme el robot avanza en el entorno. Actualmente, existen diferentes algoritmos para obtener la separación de un entorno en nodos, pero las soluciones efectivas se realizan de manera 'off-line', es decir, a posteriori una vez se tienen todas las imágenes captadas. El trabajo presentado permite realizar esta tarea de manera incremental mientras el robot

explora el nuevo entorno. Realizar esta tarea mientras se visita el resto del entorno puede ser muy interesante ya que tener el mapa separado por nodos con relaciones de proximidad entre ellos se puede ir utilizando para tareas de localización jerárquica. Además, es posible reconocer entornos ya visitados o similares a nodos pasados.

Por último, la tesis también incluye el estudio de técnicas de aprendizaje profundo (‘deep learning’) para tareas de localización. En concreto, se estudia el uso de las redes siamesas, una técnica poco explorada en robótica móvil, que está basada en las clásicas redes convolucionales, pero en la que dos imágenes son evaluadas al mismo tiempo. Estas redes dan un valor de similitud entre el par de imágenes de entrada, lo que permite realizar tareas de localización visual. En este trabajo se expone esta técnica, se presentan las estructuras que pueden tener estas redes y los resultados tras la experimentación. Se evalúa la tarea de localización en entornos heterogéneos en los que el principal problema viene dado por cambios en la iluminación de la escena. Con las redes siamesas se trata de resolver el problema de estimación de estancia, el problema de localización jerárquica y el de localización absoluta.

Agradecimientos

En primer lugar, me gustaría dar las gracias a mis directores de tesis, Óscar Reinoso y Luis Payá. Habéis sido un ejemplo docente e investigador. En unos meses pasé de disfrutar el cómo dabais clases a trabajar para vuestro grupo de investigación, sin apenas saber dónde me estaba metiendo. Creedme, ahora que veo que mi destino se acerca más a la docencia muchas veces pienso: ¿Cómo explicaría Óscar esto? ¿Qué ejemplo pondría Luis para que la gente se enterara? Muchas gracias por las lecciones que me habéis dado, por las horas que me habéis dedicado y por confiar en mí.

En segundo lugar, me gustaría agradecer a mis compañeros de fatiga. Sin Sergio Cebollada ni María Flores este trabajo tampoco hubiera sido posible. Sergio, has sido un verdadero compañero, siempre atento y dispuesto a colaborar, a ayudar a los demás y a crear un buen entorno. Tienes un don para guiar y sacar potencial de otras personas, sigue aprovechando esto. Quién nos iba a decir que una relación que empezó explorando las imágenes omnidireccionales iba a traspasar la universidad y nos haya llevado a tantos sitios. Por otro lado, ¡María, tú siempre en mi equipo! Creas muy buen ambiente, eres atenta, ayudas a los demás y nos escuchas y aconsejas. Me quedo con esas tardes en las que no nos salía nada, nos mirábamos y decíamos 'Creo que no valemos para esto, aquí hay gente mejor que nosotros'. Que el síndrome del impostor no nos devore nunca y nos demos cuenta que si nos lo proponemos somos capaces de mucho.

Además, también me gustaría agradecer a otros profesores que me habéis acompañado por este grupo ARVC. Luis Miguel J., Arturo G., David V., Mónica B., Adrián P., gracias por las lecciones y consejos que me habéis aportado. A otros compañeros que he tenido: Juanjo, Julio y Yerai, tenéis un potencial inmenso, me hubiera gustado coincidir más y aprender con vosotros.

También me gustaría dar las gracias a Amalia y Orlando, amistades que empezaron en la Universidad y que han pasado a ser gente cercana e importante durante este camino.

Tampoco puedo olvidarme del Prof. Dr, Kirill Krinkin, gracias por acogerme para una estancia en la Universidad de Electrónica 'LETI' de San Petesburgo y del profesor Prof. Dr. Hubert P. H. Shum por permitirme realizar la colaboración en la Universidad de Durham. Ambas estancias me han permitido crecer como investigador y me han ayudado a ser profesional que ahora soy.

Fuera del ámbito universitario, quiero agradecer a mis padres: Gertru y Vicente, por la educación recibida y permitirme vivir en una situación privilegiada en la que he podido dedicarme a estudiar y trabajar en lo que me ha ido gustando. La elaboración de esta tesis ha coincidido con una etapa de difícil pero juntos la hemos ido llevando lo mejor que hemos sabido. A mi hermano Alberto, gracias por acompañarme durante estos años, espero estar siendo un ejemplo positivo hacia ti como tantas veces lo has sido tu para mí. También quería dar las gracias a María José, por aparecer por la familia y enseñarme valores como el compromiso y la constancia. Al resto de familiares que os habéis interesado por

mí durante esta etapa, aunque me cueste explicaros en que he estado trabajando, me habéis apoyado mucho.

También me quiero acordar de personas que me han acompañado durante estos años. Даша, спасибо за твою поддержку!. No quiero olvidarme de amistades que me han acompañado durante estos años. La familia que se elige. Sois muchos a los que tengo que agradecer el apoyo y la oportunidad que me dabais para despejarme cuando el trabajo empezaba a absorber.

Por último, dar las gracias a Alba. Tu aparición ha sido el empujón para terminar este proyecto. Gracias por como eres, por todo lo que me aportas y enseñas cada día.

En general, quiero agradecer a todas las personas con las que he coincidido y que han contribuido en esta etapa de mi vida.

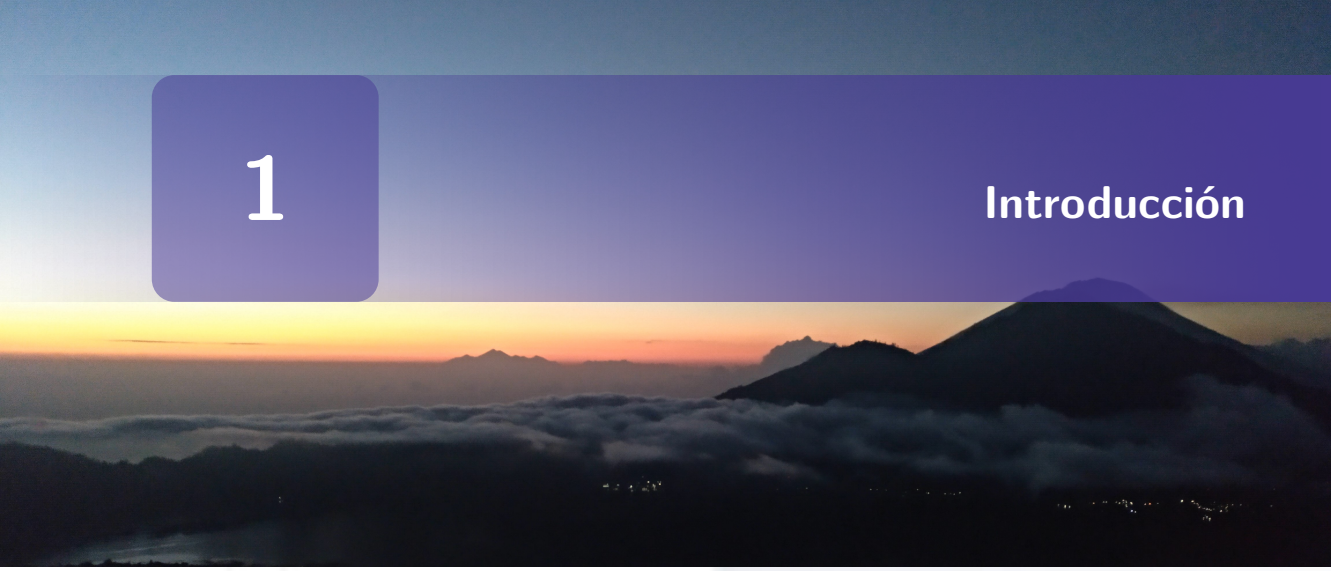
Esta tesis ha sido cofinanciada por la Unión europea a través del Programa Operativo del Fondo Social Europeo (FSE) de la Comunitat Valenciana 2014-2020 a través de la beca predoctotal ACIF/2018/224.

“Viure, viure no és estar vius.
Viure és l’actitud d’omplir la vida.”

Índice general

Índice de cuadros

Índice de figuras



1 Introducción

1.1. Motivación

La tesis que se presenta tiene como idea principal el avance en el campo de la robótica móvil, un tema de rabiosa actualidad que se ha visto mejorado durante los últimos años gracias al desarrollo de la tecnología. El campo de la robótica se ve desarrollado como complemento a la vida humana. Los robots son sistemas capaces de reemplazar o complementar a un humano en tareas mecánicas, en rutinas repetitivas y/o en tareas peligrosas. Para desarrollar estos trabajos, los robots necesitan cierto grado de autonomía. En el caso de los robots autónomos, es necesario desarrollar un sistema de navegación valido para cualquier entorno sin la ayuda un operador humano. Para conseguir este sistema de navegación, un robot autónomo requiere abordar la tarea de creación de mapas, que consiste en obtener información de un entorno desconocido y crear un modelo que represente el entorno, y la tarea de localización, que consiste en conocer la posición y orientación en la que el robot se encuentra respecto del modelo creado. Estas tareas deben ser realizadas de manera suficientemente precisa y con un coste computacional relativamente bajo. Para llevar a cabo estas tareas con una precisión adecuada es necesario el uso de sensores. Muchos sensores se han estudiado para las tareas de navegación. Por ejemplo, Bloesch *et al.* [22] proponen el uso de información cinemática adquirida por un encoder y una IMU (Unidad de Medida de Inercia), Kim *et al.* [78] proponen el uso de información de odometría y GPS junto al filtro extendido de Kalman para estimar la localización del robot. Lingemann *et al.* [92] proponen una aproximación basada en láseres para tareas de tracking conociendo así la pose del robot móvil. Por último, también se han presentado soluciones que utilizan la información que proviene de láseres y odometría para la creación de mapas [37].

No obstante, estos sensores no hacen uso de la gran cantidad de información visual que puede ser de gran ayuda para las tareas de localización y creación de mapas. Para ayudarse de la información visual se hace uso de los sensores de visión o cámaras. Los sensores de visión han sido utilizados con resultados que muestran su eficacia en tareas de creación de mapas y localización, así como en ensayos de navegación robótica. Por ejemplo, Häne *et al.* [59] usan cámaras de ojo de pez para evitar obstáculos, crear mapas 3D y realizar una localización visual, Pire *et al.* [133] proponen un sistema SLAM (localización y creación de mapas de manera simultánea) en tiempo real y para ello utilizan un para de cámaras estéreo y Arth *et al.* [11] proponen un sistema SLAM para exterior basado en la herramienta de mapas 2.5D. Fuentes-Pacheco *et al.* [51] en 2015 presentan un estado del arte sobre algoritmos de SLAM visual y Cebollada *et al.* [35] 2020 presentan un estado del arte sobre herramientas de navegación autónoma basada en técnicas de visión.

A pesar de lo investigado, estos estudios presentan debilidades en ciertas condiciones del entorno. Los sistemas son muy sensibles ante cambios en las condiciones de iluminación. Por ejemplo, un robot puede crear de forma autónoma un mapa un día soleado pero localizarse en él con otras condiciones de iluminación puede ser complicado o un robot que trabaje en entornos de subsuelo deberá realizar sus operaciones sin apenas luz y lidiando con muchas sombras. Adicionalmente, los sistemas serán sensibles ante otras condiciones dinámicas del entorno, como pueden ser cambios en el mobiliario, actividad humana, oclusiones o ruido visual entre otras. El estudio de cómo afrontar y mitigar estos efectos es una importante linea a seguir. Conseguir que este tipo de perturbaciones afecten mínimamente en las tareas de navegación supondría una revolución en el mundo de la robótica autónoma.

Una vez se considera trabajar con información visual, diferentes puntos de vista se han llevado a cabo en los últimos estudios. Por un lado, muchos autores han investigado soluciones utilizando cámaras monoculares [121]. Otros autores se decantan por configuraciones binoculares [184] [57] o trinoculares [70].

El problema asociado a estas configuraciones es el campo de visión. Los sistemas expuestos tienen un campo visual relativamente pequeño y delimitado. Por ello, para captar información de todo el entorno se debe capturar diferentes imágenes. Como alternativa a estas configuraciones se encuentran las cámaras omnidireccionales. Estas cámaras proporcionan un campo de visión de 360 grados en el plano por el que se mueve el robot. Gracias a estos sistemas es posible desarrollar algoritmos que optimicen esta característica. Con cámaras con un campo de visión de 360 grados las características de la imagen permanecen mas tiempo visibles y son más estables. Las mismas características se visualizan con cualquier orientación del robot y desarrollando los algoritmos adecuados se puede resolver tanto la posición como la orientación relativa.

En la literatura se puede encontrar diferentes estudios sobre el uso de las cámaras omnidirecionales para tareas de creación de mapas, localización y navegación de robots móviles [176],[19], [171], [116], [108],[33]. Payá *et al.* [128] realizan un estudio exhaustivo del uso de estas cámaras y presentan un estado del arte de los algoritmos de localización y creación de mapas que hacen uso de la información visual obtenida por

cámaras omnidireccionales. La figura 1.1(a) muestra un ejemplo de un robot móvil con una cámara omnidireccional montada sobre él y la figura 1.1(b) muestra un ejemplo de imagen omnidireccional.

Las tareas de localización y creación de mapas con información visual han tomado dos perspectivas de trabajo tradicionalmente. Por un lado, se puede trabajar con información local; detectando y describiendo información de puntos característicos. Por otro lado, se trabaja con los los algoritmos holísticos o de descripción global, con los que se construye un único descriptor de toda la imagen. En los últimos años también se ha trabajado con técnicas de 'deep learning' para estas tareas. Esta técnica se asemejaría más a la segunda perspectiva ya que el objetivo es describir una imagen con un vector.

(a) (b)

Figura 1.1: (a) Ejemplo de un Pioneer P3-AT ® , robot equipado con un sistema de visión omnidireccional y un sensor láser. (b) Ejemplo de una imagen omnidireccional capturada en un entorno de exterior. Imagen obtenida de la base de datos Fukuoka Datasets [106].

Respecto a la tarea de creación de mapas se han seguido dos vertientes principales a la hora de la creación: por un lado, en la literatura relativa al mapping se pueden encontrar los mapas métricos; estos representan el entorno con precisión métrica. Por otro lado encontramos los mapas topológicos que representan la realidad con una serie de grafos que representan la conectividad entre nodos. En relación a ambas opciones, también se puede encontrar soluciones que tratan de construir un mapa jerárquico. En los mapas jerárquicos unas primeras capas permiten una localización aproximada mientras que capas más profundas realizan una aproximación más precisa a la solución. Teniendo en cuenta los problemas de localización y creación de mapas, los mapas jerárquicos pueden suponer una alternativa eficiente para la navegación ya que pueden

proporcionar una solución precisa con un menor tiempo de computación [125], [161], [58] y [68].

En relación con la anterior información, el modo de construcción de los mapas también supone un punto importante de estudio. Para la construcción de mapas es habitual el uso de técnicas de clustering espectral. Esto supone ciertos inconvenientes como la necesidad de conocer todas las entidades y el número de nodos previamente, lo que supone la imposibilidad de crear el mapa de manera *on-line* mientras se explora el entorno. Autores como Valgren *et al.* [175] estudian la posibilidad de construir mapas de manera incremental. Este problema ha sido abordado en este trabajo con el objetivo de conseguir crear y mantener mapas con información visual y descriptores globales.

Por último, las técnicas de *machine learning* también han ayudado a resolver problemas relacionados con la robótica móvil en los últimos años. Por ejemplo, Gonzalez *et al.* [54] ha usado estas técnicas para el movimiento de misiones robóticas en Marte. Dymczyk *et al.* [48] proponen el uso de una red para clasificar observaciones y llevar a cabo la tarea de localización de una manera más robusta. Cebollada *et al.* [36] realizan un estudio comparado entre técnicas clásicas de apariencia global y técnicas de *machine learning* en creación de mapas y localización ante cambios de iluminación en la escena.

El propósito de esta tesis es prolongar el uso de técnicas de 'machine learning' para tareas de navegación autónoma de robot móviles. Para ello se introduce el estudio de redes siamesas [88] para las tareas de localización y mapping. Se propone un estudio de este tipo de redes neuronales para tareas de localización jerárquica del entorno, realizando un estudio de comparación entre la tarea de localización por capas y la de localización global del entorno.

1.2. Objetivos

El objetivo principal de la tesis es la optimización de las tareas de localización y construcción de mapas de manera incremental haciendo uso de cámaras omnidireccionales y descriptores de apariencia global. Para la optimización de estros trabajos se busca reducir el error de localización y el coste computacional. Para llegar a estos objetivos se establecen los siguientes propósitos de estudio:

- **Comparación entre métodos de apariencia global para la descripción de imágenes.** Para la tarea de localización basada en información visual se utilizan técnicas de compresión las cuales se han evaluado en este trabajo. El objetivo es realizar un estudio de estas técnicas, los parámetros de los cuales dependen y cómo les afectan los efectos visuales adversos en su precisión ante tareas de localización.
- **Tarea de creación de mapas de manera incremental.** Para la tarea de mapping se pretende poder realizar esta construcción de manera incremental y *on-line*. Se estudia el uso de descriptores de apariencia global y de los parámetros que pueden influir notoriamente en el algoritmo de creación de mapas.

- **Tarea de mantenimiento de mapas** Asimismo, se pretende realizar un estudio del mantenimiento de los mapas una vez han sido creados y el robot vuelve a pasar por una estancia previa pero nueva información es detectada.
- **Estudio de las redes Siamesas para tareas de robótica móvil.** Esta tesis también pretende estudiar el uso de técnicas de 'machine learning' como las redes neuronales para tareas de navegación. Se realiza un estudio de redes neuronales convolucionales conocidas pero también otras técnicas emergentes como las redes siamesas. Esas redes tienen dos entradas y como salida se obtiene un valor de similitud entre ambas imágenes.
- **Evaluación del uso de redes para comparar tareas de localización jerárquica y localización global.** Una vez estudiada la posibilidad de usar redes para tareas de navegación, se realiza una evaluación de las tareas de localización jerárquica en la que por capas se afina la posición y una localización global en un mapa topológico. Esta comparación evalúa diferentes descriptores y redes.

1.3. Marco de la tesis

La presente tesis doctoral se ha desarrollado bajo un marco financiado por diferentes colaboraciones, becas y proyectos de investigación.

1.3.1. Becas y premios

Los trabajos de la tesis han sido posibles gracias al apoyo de la beca ACIF de la Conselleria de Educación, Investigación, Cultura y Deporte cofinanciada por la Unión Europea a través del Programa Operativo del Fondo Social Europeo (European Social Fund, ESF). Esta beca, cuyo número de referencia es ACIF/2018/224, ha financiado al autor principal de la tesis durante tres años (de noviembre de 2018 a septiembre de 2021). Esto ha proporcionado la posibilidad de trabajar en la investigación durante este periodo.

Además, otras becas y programas han permitido al autor realizar cortas estancias en universiadades del extranjero. Estas estancias están descritas en el siguiente apartado. Por un lado, una primera estancia en la Universidad de Electrónica 'LETI' de San Petesburgo (Rusia) y se realiza con la financiación de la Universidad Miguel Hernandez de Elche. Por otro lado, una segunda estancia se realiza en la Universidad de Durham (Reino Unido) gracias la subvención para proyectos de I+D+i desarrollados por grupos de investigación emergentes financiado por la Generalitat Valenciana.

1.3.2. Estancias de investigación y colaboraciones

De Agosto a Noviembre de 2019, el autor de esta tesis realizó una estancia de colaboración durante tres meses en el Departamento de software y aplicaciones por ordenador (Department of Software Engineering and Computer Applications) de la Universidad de Electrónica 'LETI' de San Petesburgo (Saint Petersburg Electrotechnical University 'LETI') en Rusia. El objetivo de esta estancia de investigación fue

colaborar con Prof. Dr. Kirill Krinkin, que investigaba sobre el uso de descriptores de apariencia local para una rápida localización, utilizando estos estudios principalmente para la navegación de drones. Esta colaboración permitió al doctorando el trabajar con otro tipo de técnicas en la localización con información visual y la profundización en la programación con Python. Estos estudios han sido financiados por la Universidad Miguel Hernández de Elche (Miguel Hernandez University).

Además, de Octubre a Diciembre de 2020, el autor de esta tesis realiza una colaboración de dos meses en el Departamento de computación y ciencias de la informática (Department of Computer and Information Sciences) de la Universidad de Durham, Reino Unido. El objetivo de esta colaboración con el Prof. Dr. Hubert P. H. Shum fue investigar el uso de nuevas técnicas de inteligencia artificial para diversas tareas. Entre ellas la investigación de creación de mapas 3D o localización robótica con redes neuronales emergentes. Esta estancia de colaboración se ha financiado gracias a la Generalitat Valenciana y la subvención para proyectos de I+D+i desarrollados por grupos de investigación emergentes.

1.3.3. Proyectos de investigación

Durante el desarrollo de la tesis, el autor ha participado en diferentes proyectos de investigación que se detallan a continuación.

- **"EMERG2020. Subvención para proyectos de I+D+i desarrollados por grupos de investigación emergentes".**Subvención de la Generalitat Valenciana desde Enero 2020 hasta Diciembre 2020.
- **"Creación de modelos jerárquicos y localización robusta de robots móviles en entornos sociales".** Proyecto financiado por la Generalitat Valenciana desde enero 2019 hasta diciembre 2020. AICO/2019/031. Subvenciones para grupos de investigación consolidables.
- **"Creación de Mapas Mediante Métodos de Apariencia Visual para la Navegación de Robots ".** Proyecto financiado por el Ministerio de Ciencia e Innovación desde enero 2017 hasta diciembre 2019. DPI2016-78361-R. Ayudas a proyectos de I+D+I. Programa estatal de investigación, desarrollo e innovación orientada a los restos de la sociedad.

1.4. Publicaciones

1.4.1. Publicaciones en revistas indexadas JCR

Las principales contribuciones son dos artículos publicados en revistas indexadas JCR. Uno de ellos en la primera categoría de cuartil (Q1) y otro categorizado en el segundo cuartil (Q2). Estas publicaciones están en la linea de los propositos de la tesis y se describen a continuación.

- The Role of Global Appearance of Omnidirectional Images in Relative Distance and Orientation Retrieval. [146]
  V. Román, L. Payá, A. Peidró, M. Ballesta, O. Reinoso
  Sensors. 21(10), 3327 (Mayo 2021)
  ISSN:1424-8220. Ed. MDPI
  JCR-SCI Impact Factor(2021, último dato disponible en el momento de la aceptación del artículo): **3.576**, Quartile **Q1**
  Web: https://doi.org/10.3390/s21103327
  DOI: 10.3390/s21103327

- Creating Incremental Models of Indoor Environments through Omnidirectional Imaging. [142]
  V. Román, L. Payá, S. Cebollada, O. Reinoso
  Applied Sciences. Vol 10(18),6480 (Septiembre 2020)
  ISSN:2076-3417. Ed. MDPI
  JCR-SCI Impact Factor(2020, último dato disponible en el momento de la aceptación del artículo): **2.679**, Quartile **Q2**
  Web: https://doi.org/10.3390/app10186480
  DOI: 10.3390/app10186480

Estos artículos están incluidos en el apéndice Apéndice A.

1.4.2. Trabajos derivados de la Tesis Doctoral presentados en congresos de investigación

En este apartado se presentan otros trabajos desarrollados durante la elaboración de la tesis que han sido presentados y publicados en congresos y otras revistas, consecuencia de las tareas acometidas y las colaboraciones realizadas en el marco de la presente tesis y en las diferentes líneas de investigación llevadas a cabo

- V. Román, L. Payá, S. Cebollada, A. Peidró, O. Reinoso. Evaluating the Robustness of New Holistic Description Methods in Position Estimation of Mobile Robots. Lecture Notes in Electrical Engineering. Informatics in Control, Automation and Robotics (2022) - 793 (207-225). Ed. Springer [141].
- V. Román, L. Payá, S. Cebollada, A. Peidró, O. Reinoso. An Evaluation of New Global Appearance Descriptor Techniques for Visual Localization in Mobile Robots under Changing Lighting Conditions. ICINCO 2020: 17th Intl. Conf. On Informatics in Control, Automation and Robotics (Streaming Online, 7-9 julio 2020). Ed. INSTICC [144].
- V. Román, L. Payá, M. Flores, S. Cebollada, O. Reinoso. Performance of New Global Appearance Description Methods in Localization of Mobile Robots. Robot 2019, Advances in Intelligent Systems and Computing (2019) - 2 (351-363) Ed. Springer [145].

- V. Román, L. Payá, O. Reinoso. Evaluating the robustness of global appearance descriptors in a visual localization task, under changing lighting conditions. ICINCO 2018, 15th International Conference on Informatics in Control, Automation and Robotics (Oporto, Portugal, 29-31 agosto, 2018) Ed. INSTICC [147].
- V. Roman, S. Cebollada, L. Payá, M. Flores, A. Gil, O. Reinoso. Evaluación de nuevos modos de empleo de los descriptors de apariencia global en tareas de localización. XL JORNADAS DE AUTOMATICA (El Ferrol (Spain), 4-6 September 2019). Ed. CEA-IFAC ISBN:978-84-9749-716-9. pp. 842-848 [140].
- V. Roman, L. Paya, A. Peidro, D. Valiente, L.M. Jimenez, O. Reinoso. Evaluación de descriptores de apariencia global en tareas de localización bajo cambios de iluminacion.XXXIX Jornadas de Automatica (Badajoz (SPAIN), 5-7 Septiembre 2018)- pp. 306-313 Ed. CEA-IFAC [143].

Adicionalmente, algunos de los resultados presentados en esta tesis doctoral están pendientes de publicación. Se pretende tener publicados estos resultados próximamente en una revista indexada JCR de un alto valor de impacto. Estos resultados se presentan en el capítulo 5 y son relativos a las tareas de localización jerárquica realizadas con la herramienta de *deep learning*. En concreto se estudia el uso de las redes siamesas, una técnica bastante novedosa para abordar tareas de navegación robótica.

1.5. Estructura de esta tesis

Este documento sigue la siguiente organización:

- Capítulo 2: Este capítulo presenta un conjunto de referencias a trabajos relacionados con la robótica autónoma móvil, el tratamiento de información visual y estrategias de 'maching learning'. Se presentan los trabajos más destacados en los campos de la resolución de tareas de localización y creación de mapas, el uso de sensores en robótica, la descripción de imágenes y las herramientas de inteligencia artificial para tareas de navegación. Tras la revisión de diversa literatura de los últimos años, se presentan las ventajas y fortalezas de estos métodos y las posibles debilidades tras usar estas técnicas.
- Capítulo 3: Presenta un profundo estudio de los métodos de descripción de imágenes. En concreto, se estudian 5 métodos de descripción global con los que se describe la información de una imagen en un único vector. Para esta evaluación se estudian diferentes parámetros que pueden jugar un papel importante en esta tarea. En este sentido, el estudio considera los diferentes descriptores, diferentes parámetros asociados a cada descriptor, diferentes efectos visuales que pueden ocurrir en condiciones de trabajo reales o diferentes medidas para calcular la similitud entre descriptores. Además, este estudio muestra una evaluación del error en posición, del error en estimación de la orientación y el tiempo de computación asociado.

- Capítulo 4: Esta sección lleva a cabo la exposición de los resultados obtenidos tras evaluar un nuevo método de mapping incremental. Este nuevo algoritmo está creado expresamente para trabajar con imágenes omnidireccionales y para construir el modelo a medida que la nueva información es captada y almacenada. Se presenta un estudio tras variar condiciones, entornos a visitar y parámetros que pueden afectar a esta tarea. Además, se muestran modificaciones al algoritmo original que pueden suponer mejoras para entornos de muy largo recorrido. También se proponen técnicas para mantener el mapa creado si el robot revisita la zona y el entorno es visualmente diferente.

- Capítulo 5: Este capítulo presenta el uso de herramientas de *deep learning* para tareas de localización. En concreto, se presentan las redes siamesas, una técnica poco explorada en localización de robots móviles, que utiliza redes neuronales convolucionales CNNs (Convolutional Neural Network) adaptadas a recibir dos imágenes de entrada. Estas dos imágenes son estudiadas por la red y se obtiene un vector de cada una de las imágenes en las que se condensa la información global de la imagen. Estos vectores se comparan y se obtiene un valor de similitud entre las imágenes como salida de la red siamesa. Se han utilizado estas redes para solventar tres problemas típicos en la localización de robots autónomos: el problema de identificación de estancia, el problema de localización jerárquico y el problema de localización absoluta. Durante el capítulo se presentan diferentes arquitecturas y se evalúan diferentes parámetros que influyen en el comportamiento de la red. Posteriormente, se hace una evaluación de los resultados obtenidos para los tres problemas de localización.

1.6. Resumen de materiales, métodos y discusión de los resultados

En este apartado se presenta un resumen de los principales materiales y métodos utilizados para desarrollar las líneas de investigación presentadas en la presente tesis. Adicionalmente, esta sección presenta una breve discusión sobre los principales resultados obtenidos en cada capítulo.

1.6.1. Materiales

La siguiente lista detalla los materiales utilizados para llevar a cabo los experimentos de investigación:

- Diferentes cámaras CCD: DFK-21BF04 y DFK-41BF02.

- Espejo hiperbólico: Eizho Wide 70.

- Base de datos visual con imágenes omnidirecionales y panorámicas capturada en la Universidad Miguel Hernández [126].

- Base de datos visual con imágenes omnidirectionales capturadas bajo diferentes condiciones de iluminación capturada en entornos de interior de edificios ubicados en tres universidades: Universidad de Friburgo, Alemania; Univerdad de Ljubljana, Eslovenia; y el Centro de Inteligencia Artificial de Saarbrücken, Alemania [135].
- PC con CPU Intel Core i7-7700 ® a 3.6 GHz con GPU NVIDIA GEFORCE GTX 1080TI ®.

1.6.2. Métodos

La siguiente lista detalla los métodos desarrollados para llevar a cabo los trabajos de investigación:

- Herramientas matemáticas: Firma de Fourier, descriptor HOG, descriptor *gist*, descriptor Wi-SURF, descriptor BRIEF-gist, distancia Euclidea, distancia coseno, distancia correlation, distancia Manhattan, algoritmo de clustering espectral.
- Herramientas de 'machine learning' y 'deep learning'. Redes neuronales usadas para obtener descriptores holísticos para resolver tareas de navegación. 'Transfer learning' para adaptar arquitecturas existentes a la tarea de localización y evaluación de las redes resultantes.
- 'Data augmentation': Técnicas que permiten aumentar significativamente la diversidad de datos disponibles para el entrenamiento de las redes, sin necesidad de adquirir nuevas imágenes. En este caso, se ha desarrollado nuevas técnicas de aumento de información aplicando efectos de cambios globales y locales en la iluminación, ruido gausiano, oclusiones, rotación en las imágenes, efectos de desenfoque etc.

1.6.3. Resultados y discusión

Finalmente, esta sección resume los resultados obtenidos en os trabajos de investigación descritos en cada capítulo. Estos resultados se han publicado en congresos nacionales e internacionales o en revistas indexadas en JCR. Estas publicaciones se pueden encontrar en el apéndice A.

- Capítulo 3: Este capítulo se ha centrado en el estudio del problema de localización utilizando un modelo visual previamente construido que represente el entorno. El problema se ha resuelto como una tarea de localización visual absoluta en la que existe un modelo del entorno y la nueva imagen adquirida trata de encontrar su posición más probable de entre las imágenes del modelo. Haciendo uso de un sensor de visión catadióptrico montado en el robot móvil se estima la posición y orientación del robot. Para extraer información relevante de las imágenes se hace uso de los descriptores de apariencia global, los cuales proporcionan un único vector que describe toda la imagen. Se realiza una evaluación comparativa de seis familias de descriptores de apariencia global (Fourier Signature, HOG, *gist*,

Wi-SURF, BRIEF-Gist y Transformada de Radon). Además, se tienen en cuenta distintos efectos negativos que pueden ocurrir en tareas de localización reales como puede ser la aparición de ruido en las imágenes o de oclusiones que no permitan ver por completo la escena, realizando el estudio en condiciones ideales y cuando estos efectos aparecen en la captura de información.

- Capítulo 4: Creación de modelos de entornos interiores de manera incremental: En este capítulo se presenta un método y sus mejoras para crear mapas en entornos de interior de manera incremental, actualizando el modelo cada vez que una nueva imagen es tomada. El marco está basado en el desarrollo de un algoritmo de clustering incremental que actualiza el mapa cada vez que se adquiere una imagen o se tiene en cuenta un nuevo evento como la creación de nuevos nodos o fusión de nodos existentes.. Los experimentos se han realizado en entornos interiores por los que el robot ha navegado bajo condiciones reales, incluyendo variación de iluminación y cambios introducidos por la actividad humana. Los robots con los que se han realizado los experimentos están equipados con un sistema de visión omnidireccional y la única información usada para construir el mapa jerárquico son las imágenes capturadas por el sistema. Para describir las imágenes se hace uso de dos métodos de descripción global de las imágenes: HOG y *gist*. La sección de experimentos muestra el rendimiento del algoritmo y la influencia de los principales parámetros en el resultado final. Además, se muestra una comparativa para evaluar los resultados obtenidos por el algoritmo propuesto con un algoritmo de clustering espectral que funciona *off-line* y de manera absoluta (es necesario que todas las imágenes estén disponibles de antemano).

- Capítulo 5: Uso de redes siamesas para tareas de localización jerárquica y absoluta: La finalidad de este capítulo se centra en la resolución de la tarea de localización haciendo uso del *deep learning* y las redes neuronales, en especial en el estudio de las redes siamesas. Las redes siamesas permiten comparar simutaneamente dos imágenes, devolviendo un valor de similitud entre ambas imágenes. Se ha realizado una evaluación de las redes siamesas para la resolución de tres supuestos: la estimación de habitación (tarea de *room retrieval*), la localización jerárquica y la estimación absoluta de la posición. En primer lugar, en la tarea de *room retrieval* se ha tratado de encontrar la habitación más probable en la que el robot está ubicado. En segundo lugar, en la tarea de localización jerárquica el robot trata de ubicarse dentro de la estancia elegida como más probable; en este caso el robot solo comparará la nueva imagen con las imágenes almacenadas en el modelo de la estancia estimada. Por último, en la tarea de localización absoluta la nueva imagen se compara con todas las imágenes del modelo.

  Para la obtención de información del entorno se ha utilizado un sensor de visión catadióptico montado sobre un robot móvil. Para extraer información relevante de las imágenes se hace uso de las redes siamesas. Como se ha introducido, estas redes tienen la particularidad de comparar dos imágenes de entrada, de cada imagen se obtiene un vector global que define la imagen y estos se comparan dando lugar a un coeficiente de similitud entre imágenes. Además, también se

han estudiado técnicas de *data augmentation* para generar imágenes artificiales con las que entrenar la red.

2 Estado del arte

Durante los últimos años, la presencia de los robots ha aumentado substancialmente en todos los ambientes de la vida. Es habitual encontrar el uso de la robótica para llevar a cabo tareas repetitivas y específicas, así como tareas en entornos peligrosos o con resultados que deben ser muy precisos. En esta línea encontramos trabajos como [16] que propone el uso de robots para realizar tareas de bombero, [96] o [56] que proponen una serie de trabajos en relación a robots trepadores para inspeccionar muros verticales, los trabajos en relación al sistema de cirugía Da Vinci [23, 84, 75] o estudios como [136] donde se trabaja con robots para la inspección de líneas eléctricas de alta tensión. A diferencia de los trabajos anteriores, la presente tesis doctoral se va a centrar en el estudio de robots móviles autónomos, es decir, aquellos robots diseñados para navegar por el entorno sin asistencia de un operario o ayuda humana [157, 4]. Este tipo de robots puede abarcar diferentes tipos de aplicaciones, entre otras exploración de entornos peligrosos o ambientes extremos, vigilancia, reconocimiento, automatización industrial, entretenimiento, guía de museos, transporte, cuidados médicos etc. [4, 149].

Para poder realizar estas aplicaciones, el robot debe ser capaz de navegar por un entorno por sí mismo. Esto requiere un dominio de las tareas de localización y creación de mapas. Por un lado, es preciso que el robot sea capaz de elaborar un mapa del entorno con la información captada por los sensores (tarea de creación de mapas). A su vez, debe ser capaz de con un mapa previamente creado localizarse en el entorno, es decir, conocer su posición y orientación respecto de la información del mapa (tarea de localización). Las tareas de mapping, localización y navegación son cruciales para poder construir aplicaciones con robots móviles autónomos. En este capítulo se hará un estudio de los trabajos relacionados con estas tareas, así como de los sensores necesarios para captar la información.

El capítulo se centra en un primer momento en los sensores de visión como herramienta para captar información del entorno. A continuación, el foco se pone en las técnicas de descripción de información visual. Una vez conocidas las técnicas de descripción se expone un estudio de los trabajos relacionados con las tareas de localización, mapping y navegación integrada. Para finalizar, se expone nuevas técnicas basadas en la Inteligencia Artificial (IA) para aplicaciones de robótica móvil.

2.1. Uso de sensores de visión en robótica

Para poder realizar las tareas de navegación por el entorno, el robot debe recibir información del exterior y procesarla. De la captación de información del entorno se encargan los sensores. Dicha información permitirá que el robot resuelva las tareas de creación de mapas y localización, y otros problemas más concretos como evitar colisiones con objetos, muros o personas y evitar condiciones extremas o peligrosas para el robot. Reinoso y Payá [138] presentan un número especial sobre la actualidad en el campo de navegación autónoma y una variedad de enfoques relacionados con esta tarea. Para captar información del entorno, se pueden utilizar diferentes herramientas, como laser [177], sonars [77] o GPS [43]. Pero en lo que se refiere al estudio realizado en este trabajo la información se capta desde cámaras, por ello, este estado del arte se centra en sensores de visión. Las cámaras ofrecen mucha información del entorno con un coste relativamente bajo y pueden ser utilizadas en entornos de interior y de exterior. Muchas veces pueden aparecer como complemento a otros sensores, como la combinación del sensor sonar con información visual para realizar una tarea de SLAM (Simultaneous Localization And Mapping) [39]. Por todo ello, los sensores de visión tienen un largo recorrido para ser utilizados con propósitos de localización y mapping de robots móviles [139].

Durante los últimos años se han llevado a cabo muchos avances y esto supone una amplia variedad de soluciones. La constante evolución de las técnicas de visión para tareas de creación de mapas, localización y navegación está ampliamente documentada. Se puede encontrar documentación sobre trabajos de la década de los dosmil [24], complementada con la evolución de estas técnicas con trabajos de la última década [35].

Respecto a los sensores empleados, se puede encontrar diferentes configuraciones. Entre ellos se encontran soluciones basadas en sistemas monoculares [117, 148], binoculares (sistemas estéreo) [30, 169] y otro tipo de configuraciones más particulares [70, 103]. Aunque recientemente, han ganado popularidad los sistemas de visión omnidireccionales [86, 79]. Estos son capaces de capturar información en todas las direcciones y por tanto se le proporciona al robot un campo de visión de 360º. Los sistemas de visión omnidireccionales tienen grandes ventajas respecto a sistemas visuales con otras configuraciones de cámaras. Las imágenes obtenidas por estos sistemas tienen un campo de visión más amplio, esto permite recoger más información del entorno que otros sistemas de visión pudiendo así crear un modelo de la escena con menor cantidad de imágenes. Asimismo, el tener mayor campo de visión, supone que las características de las imágenes permanezcan durante más tiempo en el campo de

visión del robot mientras este se mueve, cualidad importante para tareas de navegación. Además, este tipo de sistemas proporcionan información suficiente para estimar la posición del robot independientemente de su orientación. También es posible estimar la orientación relativa entre dos imágenes captadas en la misma posición. Por último, las imágenes omnidireccionales contienen información suficiente de la escena para poder extraer información si durante el recorrido algún objeto o persona ocluye parte de la imagen.

Para la obtención de imágenes omnidireccionales se puede hacer uso de diferentes técnicas: es posible realizar una composición de una imagen omnidireccional con diferentes imágenes capturadas desde distintos puntos de vista o se puede obtener con una única cámara que captura el reflejo de la escena en un espejo convexo, esta técnica se denomina sistema de visión catadióptrico. Normalmente, estos sistemas están formados por una cámara monocular convencional que apunta a un espejo convexo, que dependiendo del tamaño y forma ofrecerá diferentes tipos de imágenes. La información visual proporcionada por estas cámaras se puede representar en diversos formatos, tales como omnidireccional, panorámico y vista en planta [74, 90, 21]. Durante esta tesis se hace uso de las representaciones panorámicas y omnidireccionales. Tal y como se detallará, esos formatos contienen la información necesaria para conocer la posición del robot y estimar su orientación relativa cuando el robot registra un movimiento en el plano del suelo.

2.2. Técnicas de descripción de imágenes

Como se ha comentado en este capítulo, los robots móviles tienen que solucionar las tareas de mapping y localización para realizar tareas de navegación autónomas. Para llevar a cabo estas tareas nos vamos a centrar en extraer información del entorno con sensores visuales. Pero solucionar las tareas de navegación utilizando únicamente información visual es una tarea bastante desafiante. Las imágenes contienen mucha información y se debe extraer la más relevante para afrontar estos problemas con un coste computacional adecuado. Para ello se hace uso de herramientas de descripción de información visual. Los descriptores de apariencia local son la solución adoptada más conocida y consisten en detectar y extraer información de puntos locales, landmarks o regiones de la imagen y describirla. Algunos de los descriptores de apariencia local más conocidos son SIFT (Scale-Invariant Feature Transform) [97], SURF (Speeded-Up Robust Features) [18] y BRIEF (Binary Robust Independent Elementary Features) [27]. Se trata de una solución bastante madura que muchos autores han empleado en diversos trabajos de mapping y localización. Por ejemplo, Angeli et al. [8] proponen el uso de estas técnicas para tareas de localización topológica. Valgren and Lilienthal [175] los utilizan para tareas de localización en entornos de exterior. Murillo et al. [116] presentan un estudio comparativo de tareas de localización en entornos de interior empleando diferentes descriptores de apariencia local. Una evaluación exhaustiva de este tipo de descriptores se puede encontrar en los trabajos de Gil et al. [53]. En ellos se evalúa su repetitividad, la invarianza ante pequeños cambios y la robustez de los descriptores ante diferentes condiciones de percepción. Según los diferentes estudios, los descriptores de apariencia local son una apuesta segura para muchas condiciones y tareas, pero estos

su desempeño se resiente cuando aparecen condiciones no esperadas en la captación de la imagen. Teniendo en cuanta que la información puede cambiar cuando el robot se mueve y la perspectiva cambia, y también debido a cambios en las condiciones de iluminación, oclusiones por muebles, personas u objetos y ruido durante la adquisición, es necesario calcular descriptores más robustos e invariantes. En este contexto aparecen los descriptores de apariencia global, una alternativa capaz de construir descriptores robustos ante cambios de condiciones. Los descriptores de apariencia global describen la imagen en conjunto, sin detectar landmarks ni características locales, representando una imagen con un único vector descriptor. Este método crea una representación más intuitiva del entorno que simplifica el proceso de navegación y permite la construcción de un mapa topológico del entorno. Como cada imagen se describe con un único descriptor, la tarea de localización se simplifica a una comparación simple de vectores. Además, calcular el descriptor de apariencia global de una imagen omnidireccional constituye una alternativa muy potente debido al alto campo de visión de las imágenes que permiten construir descriptores robustos e invariantes a la rotación del robot. A continuación se van a presentar algunos de los descriptores de apariencia visual utilizados hasta el momento.

En los últimos años se ha puesto en valor las técnicas de descripción global. Uno de los métodos más empleados en es el basado en el Histograma de las Orientaciones de los Gradientes (HOG). Utilizando este descriptor se han resuelto diferentes problemas dentro del campo de la robótica móvil. Por ejemplo, Dalal y Triggs [46] explican cómo usarlos para detectar peatones o Paya et al. [6, 130] usan HOG para solucionar problemas de localización jerárquica. El método *gist* es otro descriptor de apariencia global, este fue propuesto por Oliva y Torralba [122] y fue pensado para trabajar inspirado en la forma biológica de percepción para tareas de localización, fue testeado para diferentes tareas en entornos de exterior en [156], además, también fue utilizado por Zhou et al. [189] para tareas de localización relacionando nuevas imágenes con *frames* almacenados en la base de datos. Además de estos, es posible encontrar otros descriptores de apariencia global como la Transformada Discreta de Fourier [107], alternativa utilizada para tareas de creación de mapas [131], o la Transformada de Radon [137], usada para encontrar el vecino más cercano en un mapa previamente creado [19]. También es posible encontrar técnicas más recientes basadas en *Deep Learning*, estas técnicas hacen uso de la Inteligencia Artificial con la finalidad de crear descriptores globales de la imagen. Por ejemplo, Xu et al. [181], y Li et al. [89] proponen descriptores holísticos basados en CNN (Convolutional Neural Network) para encontrar la posición más probable de un robot; Ballesta et al. [17] estudian tareas de localización utilizando CNNs y capas de regresión como descriptores de apariencia global; finalmente Cebollada et al. [33] llevan a cabo una comparativa entre los descriptores de apariencia global analíticos y las nuevas técnicas basadas en IA (Inteligencia Artificial) y CNN para tareas de localización.

2.3. Tarea de localización

Para el desarrollo de la robótica autónoma es crucial que el robot sepa localizarse en el entorno por el que navega o explora. Para llevar a cabo una tarea de localización

pura, el entorno debe de estar modelado con anterioridad. En este sentido, se debe crear un mapa o un modelo previo y una vez se dispone de esta información, se puede realizar la tarea de localización. Recientemente se ha presentado gran variedad de trabajos dentro de este área. Nazemzadeh et al. [118] proponen una técnica donde se fusiona odometría y giroscopio con información de posición y altura haciendo uso de landmarks basadas en códigos QR para optimizar la tarea de localización y minimizar el error. Ming-Yi et al. [111] presentan una solución de localización multisensorial, fusionando información laser con información visual descrita con descriptores de apariencia local FAST y BRIEF. Por último, dentro de la localización utilizando información visual se puede destacar los trabajos de localización utilizando redes convolucionales (CNNs) [50].

Basándonos únicamente en las soluciones donde la tarea de localización se resuelve utilizando la descripción de imágenes, tarea más cercana a la que se propondrá en posteriores capítulos, es posible encontrar diferentes soluciones basados en la descripción local de puntos característicos [185, 99, 28, 153, 91]. También se encuentran soluciones basadas en una descripción global de la imagen como [109, 116, 156, 126, 76, 142]. Ambos enfoques se presentan de forma más profunda en el siguiente capítulo donde se expone la tarea de localización.

2.4. Métodos de creación de mapas y clustering

Otro tema importante de estudio dentro de la robótica móvil es la creación de mapas. En la literatura se puede encontrar dos estrategias para afrontar el problema de creación de mapas. Primero, se encuentran los mapas métricos, que representan el entorno con precisión geométrica. Segundo, se pueden estudiar los mapas topológicos, que describen el entorno habitualmente por medio de un grafo de conexión donde se encuentran las zonas relevantes del entorno y las conexiones entre ellas, pero no poseen información métrica. Siguiendo esta estrategia, diferentes autores proponen almacenar la información en mapas jerárquicos. Estos están compuestos de varias capas que definen del entorno con diferente precisión de información. Esta disposición permite resolver el problema de localización por fases. Primero se hace una bruta, pero rápida localización utilizando capas de alto nivel; después, se va realizando una localización cada vez más fina profundizando en el mapa jerárquico y utilizando capas de niveles más bajos. Por ello, el uso de mapas jerárquicos constituye una alternativa eficiente a la construcción de mapas con robots autónomos [125, 58, 68].

Focalizando en los mapas jerárquicos, Balaska et al. [15] desarrollan un mapa semántico no-supervisado y proponen un método de localización en él. Se utilizan puntos SURF para llevar a cabo una tarea de clustering y el mapa se corrige utilizando odometría. Korrapati y Mezouar [79] utilizan el agrupamiento de imágenes en clusters y proponen un método de cierre de bucle con información visual con el objetivo de construir mapas jerárquicos; para ello, trabajan con imágenes omnidireccionales y descriptores de apariencia local. Kostavelis et al. [80] proponen una aplicación de navegación con realidad aumentada. Dentro de esta aplicación se construye un mapa jerárquico que consiste en cuatro capas de información. En la capa más profunda se

recoge información métrica, de modo que con ella se pueden realizar de manera más precisa las tareas de localización y la navegación fina. Conforme el nivel de la capa aumenta, también lo hace el nivel de abstracción. En este sentido, en las capas de alto nivel se representa un grafo con información abstracta que representa los entornos o habitaciones de manera conceptual. El grafo en las capas de alto nivel conecta nodos con un indicador que representa la probabilidad de realizar la transición entre nodos conectados.

Adicionalmente, a la hora de la creación de mapas es imponte perfeccionar la tarea de detección de cierres de bucle ya que constituyen un paso crucial para diseñar mapas precisos, como se muestra en [83]. En este trabajo, cuando se capta una nueva imagen esta se almacena como la combinación de imágenes previas. En [29], la detección del cierre de ciclo se realiza en dos etapas: en un primer momento se usa información de los descriptores de apariencia global para encontrar candidatos al cierre de ciclo, después, el cierre de ciclo se lleva a cabo eligiendo el candidato más similar, pero esta vez utilizando características locales.

Resulta muy interesante el estudio de la construcción de estos mapas jerárquicos de manera incremental, es decir, mientras se recoge información. Los trabajos presentados hasta ahora en el ámbito de los mapas jerárquicos han construido estos mapas de forma *off-line*en su mayoría. Pero se observa atractiva la idea de contruir los mapas mientras se recoge información del entorno. Poca información se puede encontrar al respecto, pero algunos autores han planteado soluciones al problema de la creación de mapas de manera incremental. En [174], se propone un método de construcción de mapas topológicos de manera incremental. Para ello se utiliza el descriptor SIFT y el número de clusters se va incrementando continuamente mientras el robot navega. El resultado de estos estudios muestra una ruta clusterizada en nodos separados de forma correcta, pero con una cantidad de nodos que tiende a ser relativamente alta.

2.5. Navegación integrada en robótica móvil

Los conceptos de *localización* y *creación de mapas*, junto con la *planificación de trayectorias* deben ser contemplados por el proyecto si se prevé realizar un diseño de navegación integrada en robótica móvil. En la figura 2.1 se muestra cómo se relacionan estos conceptos.

Una solución donde simultáneamente se resuelve el problema de creación de mapas y el problema de localización es conocido como SLAM. Es un problema común en la robótica móvil y aparece cuando un robot realiza un recorrido por primera vez en un entorno desconocido. En este caso, el robot debe generar un modelo del entorno al mismo tiempo que se localiza en él.

Asimismo, cuando simultáneamente se realizan aplicaciones de creación de mapas combinadas con los algoritmos de planificación de trayectorias aparece la tarea de exploración. La exploración trata de crear un algoritmo que calcule la trayectoria óptima para que el robot construya un mapa del entorno lo más rápido y preciso posible.

Figura 2.1: Relación entre localización, mapping y path-planning, conceptos que engloban la navegación integrada en robótica móvil.

Por último, la combinación de las tareas de planificación de trayectorias y localización constituyen los algoritmos de localización activa. Estos algoritmos optimizan la tarea de localización y tratan que el robot recorra ciertas trayectorias que le ayuden a realizar una localización más precisa y refinada.

La combinación de los tres problemas es la denominada navegación integrada en robótica móvil, fin planteado durante el diseño de la robótica autónoma móvil.

2.6. Herramientas de inteligencia artificial para tareas de visión por computador

Para finalizar, en esta sección se presenta el uso de técnicas de inteligencia artificial (IA) para tareas en el campo de la robótica autónoma. Las tareas que puede llevar a cabo la IA son incontables. Dentro del campo de la robótica móvil las aplicaciones que hacen uso de información visual han generado un alto desarrollo en los últimos años. Entre las aplicaciones que puede tener la IA dentro de este área se puede encontrar: la detección y reconocimiento de caras [178, 67], el reconocimiento y categorización de objetos [120, 49]. De entre las posibles aplicaciones, en este apartado nos centraremos en tareas de navegación autónoma [154, 134, 124] y localización y creación de mapas [179, 65, 170].

Para las tareas de navegación usando información visual y la Inteligencia Artificial la técnica más popular es usar las Redes Neuronales Convolucionales 'Convolutional Neural Networks (CNNs)'. Actualmente se están utilizando en muchas tareas de navegación autónoma debido a su alto rendimiento en muchas aplicaciones prácticas. Están diseñadas para recibir imágenes como datos de entrada y su estructura está especialmente creada para extraer información relevante de la entrada. En este caso en que

la entrada son imágenes, su funcionamiento se basa en la búsqueda de patrones en pequeñas ventanas 2D que recorren la imagen [41]. Existen varias arquitecturas CNN bastante conocidas por su buen rendimiento y su uso es bastante extendido. Muchas de estas estructuras se suelen utilizar como base para desarrollar nuevas redes neuronales. Las más conocidas son: AlexNet [81], VGG16 [158], GoogleNet [167] o NetVLAND [9].

Durante los últimos años, las CNNs y la IA se han utilizado en robótica móvil para resolver diferentes tareas. Por ejemplo, para problemas de mapping se puede encontrar soluciones como [159] que propone el uso de una CNN para detectar información de imágenes monoculares y realizar una recolocación del robot, o [113] donde se muestra el estudio para el despliegue de un modelo para tareas de mapping del entorno que rodea a los aerogeneradores, para ello, una CNN está capacitada para extraer una estimación de la proyección de la representación del esqueleto en 3D a partir de imágenes monoculares. También, dentro de las soluciones a tareas de mapping con inteligencia artificial, se puede encontrar [113] que propone un mapping basado en una una red neuronal de regresión. Asimismo, también se han utilizado estas herramientas para tareas de localización. [168] presenta un algoritmo en tiempo real para tareas de reconocimiento de lugares utilizando CNNs que realizan tareas de localización en mapas con trayectorias (tanto en *indoor* como *outdoor*) o la utilización de *transfer learning* para llevar a cabo una localización en 6 grados de libertad utilizando la CNN de Google Iception-v4 preentrenada [160]. También se propone el uso de CNNs para predecir la existencia de objetos fuera del rango de visión de la imagen [112]. Por último, es posible encontrar otro tipo de aplicaciones de la robótica móvil empleando CNNs; dentro del campo de la navegación [188, 102] o de la tarea de SLAM (localización y mapping de forma simultánea) [98, 94].

Adicionalmente, se presentan las Redes Neuronales Siamesas. Con los años, con el avance de la investigación y el desarrollo de mejor hardware, se ha desarrollado nuevas técnicas de *Deep Learning* que pueden revolucionar la Inteligencia Artificial. Las Redes Neuronales Siamesas tienen la peculiaridad de que permiten introducir dos datos como entradas, esto permite que la red reciba dos imágenes como entrada, las compare y devuelva una salida relativa a esa comparación. El desarrollo de redes siamesas permite abordar tareas de navegación con robots móviles. Por ejemplo, Yi et al. [183] las emplean para estimar la orientación del robot o Leyva et al. [88] que utilizan las redes siamesas para reconocimiento de entornos de exterior.

Es posible encontrar más información y trabajos relativos al uso de la Inteligencia Artificial en el campo de la robótica móvil en el trabajo de Cebollada et al. [35] que presentan un estado del arte sobre estas herramientas para tareas de visión por computador.

3 Descriptores de apariencia global en tareas de localización

3.1. Introducción

Tal y como se ha descrito en los capítulos anteriores, la presencia de robots móviles se ha incrementado sustancialmente en áreas diversas como la industria, las tareas del hogar, transporte y educación, entre otras muchas. Con la mejora de sus habilidades en percepción y computación, su uso se ha incrementado y empieza a ser una herramienta útil para gran diversidad de tareas, llegando a jugar un papel imprescindible en el desarrollo de diferentes actividades. En este contexto, la creación de mapas y la localización son dos de las principales habilidades que un robot móvil debe desarrollar con cierta autonomía. El reto está en encontrar una solución para ambos problemas, teniendo en cuenta un balance entre precisión, eficiencia y robustez. Es importante que el robot sea capaz de navegar de manera autónoma y segura a través de un ambiente de trabajo real y heterogéneo [138].

En el campo de la percepción, los sensores de visión se han convertido en una solución bastante extendida para extraer información del entorno [139] debido a diferentes factores: la gran cantidad de información que son capaces de captar con un relativo bajo coste; la gran disponibilidad de obtención de los datos (el GPS por ejemplo puede no estar disponible temporalmente cuando trabaja en interior o áreas estrechas de exterior); la diversidad de configuraciones disponibles, desde una cámara simple, una cámara binocular, un sistema trinocular u otro tipo de configuraciones; y la posibilidad de combinar con otro tipo de tareas de alto nivel como la reconocimiento de personas o la detección de objetos. Entre las configuraciones disponibles, los sistemas de visión catadióptricos destacan por su gran campo de visión de 360 grados alrededor del eje de la cámara [73]. La información capturada con estos sistemas puede ser proyectada

en varias superficies, lo que permite diferentes soluciones matemáticas dependiendo de la tarea a resolver [44]. Las imágenes omnidireccionales son particularmente efectivas comparadas con las imágenes obtenidas por una cámara simple convencional ya que con una única imagen se dispone de un contexto global del entorno. De este modo, con este tipo de información, es posible conseguir descriptores de apariencia global de la escena. Los descriptores de apariencia global son una alternativa a los descriptores clásicos de apariencia local que obtenían información de regiones de la imagen y no de toda en conjunto.

Pero resolver las tareas de localización y creación de mapas utilizando únicamente información visual es un reto bastante complejo. Las imágenes contienen una gran cantidad de información y normalmente esta es redundante. Adicionalmente, esta información no solo cambia cuando el robot se desplaza, sino que también tiende a hacerlo cuando aparecen otras circunstancias como cambios de las condiciones de iluminación, ruido durante la adquisición de las imágenes u oclusiones debido a la presencia de personas, mobiliario u otros robots en el entorno. Además, cuando un robot tiene que operar en entornos de interior cabe la posibilidad de que tenga que hacer frente al *visual aliasing*, un fenómeno que hace referencia a que la información visual capturada en dos entornos distintos sea bastante similar. Teniendo en cuenta estos factores, construir un modelo visual funcional del entorno e intentar estimar la posición y orientación del robot dentro de este modelo no es una tarea sencilla y es necesario encontrar una alternativa eficiente y robusta para combatir estos fenómenos.

Para extraer información de las imágenes es posible encontrar en la literatura dos marcos principales, los descriptores de apariencia local y los de apariencia global. Esta primera familia de métodos consiste en detectar puntos o regiones de referencia y describirlos utilizando diferentes algoritmos invariantes a ciertas transformaciones, como SIFT [97], SURF [18], BRIEF [27], BRISK [85], ORB [150], FREAK [3] y LDB [182]. La segunda familia de soluciones consiste en trabajar con la imagen de manera global y tratar de construir un único descriptor por imagen, utilizando algunas soluciones como 'Principal Components Analysis' (PCA) [82], la transformada discreta de Fourier [107], los bancos de filtros de Gabor [122], los histogramas de color [173, 6], submuestrear directamente la imagen original [110] o la transformada de Radon [21]. Como es posible comprobar, habitualmente estos descriptores de apariencia global se obtienen utilizando cálculos y herramientas matemáticas, por ello también se les denomina descriptores holísticos. Pero siguiendo esta idea de cálculo de un vector que describa la imagen en conjunto es posible encontrar autores que hacen uso de herramientas de deep learning (entrenamiento profundo) para llegar a estos descriptores [164, 132].

Tradicionalmente, los investigadores se han centrado en el uso de los descriptores de apariencia local, y estos pueden ser considerados una solución madura para solventar el problema de creación de mapas y localización. Muchas soluciones han sido propuestas basadas en la utilización de estos descriptores [185, 99, 28, 153, 91]. Tradicionalmente, estos métodos requieren de la implementación de algoritmos de detección, descripción y seguimiento que tienden a ser relativamente complejos y computacionalmente lentos. Aunque han sido diseñados para ser invariantes ante diversos movimientos del robot, el comportamiento de estos descriptores puede verse deteriorado cuando otro fenómeno

inusual está presente, como cambios en las condiciones de luz, oclusiones, ruido o visual aliasing. Es posible encontrar algún análisis comparativo de este tipo de descriptores [53, 47]. Gracias a este tipo de comparativas, es posible elegir y sintonizar un método de descripción óptimo dependiendo del entorno y la aplicación.

Por otra parte, los descriptores de apariencia global se han utilizado dentro del área de robótica móvil de manera menos frecuente. Gracias a la descripción de la imagen con un único vector, es posible crear algoritmos más entendibles de manera intuitiva por el humano. La tarea de localización es más directa y se basa en la comparación de parejas de descriptores. Muchos trabajos han propuesto herramientas matemáticas analíticas adecuadas para estos problemas, como [109, 116, 156, 126, 76, 142] o utilizando la metodología de deep learning [181, 86, 35] . Estas técnicas pueden resultar muy útiles en entornos no estructurados de los que es difícil detectar landmarks. Como desventaja, estas técnicas son comúnmente utilizadas para trabajar con modelos topológicos [105, 101], ya que no es posible extraer información métrica de los descriptores de apariencia global (a no ser que se añada un sensor que permita extraer este tipo de información).

En [131] se realiza una evaluación de los métodos de apariencia global en la tarea de creación de mapas. Sin embargo, no se ha encontrado ningún trabajo que haga un estudio sistemático y profundo del rol de los descriptores de apariencia global en la tarea de localización. Por tanto, el objetivo de este trabajo es doble. Por un lado, se ha seleccionado seis métodos de apariencia global conocidos. Si ha sido necesario, los métodos se han adaptado para que puedan ser usados con imágenes omnidiecionales y obtener descriptores que contienen información útil para realizar la tarea de localización. Con este objetivo, algunos algoritmos han sido implementados para detectar la distancia y orientación relativa de manera eficiente utilizando descriptores e información puramente visual. Por otro lado, se ha realizado una comparativa de estos descriptores en la tarea de localización y se ha estudiado el comportamiento ante cambios en la pose del robot y ante otros cambios visuales del entorno. Su labor ha sido testeada y se ha estudiado la influencia de los parámetros más relevantes, completando de esta manera el trabajo presentado en [131].

El capítulo se estructurará siguiendo las siguientes secciones. La sección 3.2 presenta un estado del arte de los descriptores de apariencia global y su modo de adaptación para ser utilizados con imágenes panorámicas. Después, en la sección 3.3 se explica con detalle el modo de uso de los descriptores para realizar la tarea de localización del robot móvil. A continuación, en la sección 3.4 se presenta el material empleado, el conjunto de imágenes empleadas en los experimentos y los efectos visuales adversos analizados. El capítulo finaliza con la presentación de los experimentos y la discusión de los resultados en la sección 3.5 y con las conclusiones y posibles trabajos futuros en la sección 3.6.

3.2. Descriptores de Apariencia Global

El objetivo de esta sección es doble. Por un lado, se desea hacer un estado del arte de los métodos de apariencia global desarrollados en trabajos similares. Por otro

lado, se realiza una breve descripción matemática de los métodos revisados en la comparativa. Seis familias de descriptores globales se han elegido para ser analizados: los métodos basados en la transformada de Fourier (sección 3.2.1), en la orientación de los gradientes (sección 3.2.2), en la utilización de bancos de filtros Gabor (sección 3.2.3), en el algoritmo de descripción SURF (Speeded-Up Robust Features) (sección 3.2.4), en el algoritmo BRIEF (Binary Robust Independent Elementary Features) (sección 3.2.5) y en la transformada de Radon (sección 3.2.6). Una descripción completa de algunos de estos métodos se puede encontrar en [131, 10, 20]. Sin embargo, por favorecer el desarrollo y la claridad del capítulo, se incluye un esquema de los métodos en esta sección.

Para llevar a cabo el estudio, se considera que el movimiento del robot se realiza en el plano del suelo y las imágenes han sido capturadas utilizando en sistema de visión omnidireccional acoplado al robot. Este sistema consiste en una cámara que apunta a un espejo hiperbólico, alineados por el eje vertical. La configuración completa se presenta en la sección 3.4.

3.2.1. Descriptores basados en la Transformada Discreta de Fourier

La Transformada Discreta de Fourier(DFT) se ha utilizado por varios investigadores para extraer información relevante de la escena. Por ejemplo, Oliva y Torralba [122] proponen el uso de ventanas en las que se calcula la transformada 2D de Fourier, esto permite definir ventanas circulares para seleccionar información espacial alrededor de unos píxeles específicos. Ishiguro y Tsuji [69] proponen un enfoque alternativo para ser utilizado con imágenes panorámicas y lo denominan Firma de Fourier, Fourier Signature (FS). Menegatti et al. demuestran la efectividad de la Firma de Fourier para construir el modelo de un entorno y estimar la posición de un vehículo utilizando también el método Monte Carlo [107, 109]. Este método es útil para entornos pequeños y situaciones controladas. Adicionalmente, Stürzl et al. [162] proponen un algoritmo de en el que se utiliza la Firma de Fourier. Para poder utilizar esta altenativa previamente la imagen omnidireccional se reduce a un vector unidimensional. Horst y Möller también lo utilizan para una tarea de reconocimiento visual de entornos [66].

La Firma de Fourier (FS) permite obtener descriptores invariantes a rotaciones del robot en el plano del suelo cuando se utilizan imágenes panorámicas. Por esta razón, utilizamos esta herramienta para construir un vector basado en la Transformada de Fourier y utilizamos esta alternativa para nuestra evaluación de métodos. Este proceso de descripción se inicializa con una imagen panorámica de la escena $f(x, y) \in \mathbb{R}^{N_1 \times N_2}$. Inicialmente, la imagen se puede submuestrear para obtener un menor número de filas $k_1 < N_1$ ($k_1 = 1$ en [162]). La Firma de Fourier resultante de la escena $f(x, y) \in \mathbb{R}^{k_1 \times N_2}$ es una matriz $\mathbf{F}(u, y) \in \mathbb{C}^{k_1 \times N_2}$ que se obtiene tras calcular la Transformada de Fourier unidimensional de cada fila de la imagen. En el dominio de la frecuencia, la información más relevante se concentra en las filas con componentes de frecuencia baja, y las componentes con una frecuencia mayor tienden a tener información irrelevante y estar contaminadas con posible ruido de la imagen original. Teniendo este factor en

cuenta se pueden retener las primeras k_2 columnas y descartar el resto y realizar de esta manera un ejercicio de compresión de información. La nueva matriz de valores complejos dependerá del número de filas k_1 y de columnas k_2, y puede ser expresado con una matriz de módulos $\mathbf{A}(u, y) = \|\mathbf{F}(u, v)\|$ y una de argumentos $\Phi(u, y)$.

Basándose en la propiedad de traslación de la Transformada de Fourier unidimensional, cuando dos imágenes panorámicas han sido capturadas en el mismo punto, pero con diferente orientación alrededor del eje vertical, ambas imágenes tienen la misma matriz de módulos y matrices de argumentos diferentes. De este modo, la matriz de argumentos puede ser utilizada para calcular la orientación relativa del robot. Gracias a esta propiedad, la matriz de módulos $\mathbf{A}(u, y) = \|\mathbf{F}(u, y)\|$ puede ser utilizada como un descriptor de la posición del robot (ya que es invariante ante rotación) y la matriz de argumentos $\Phi(u, y)$ se puede utilizar como descriptor de orientación. La estimación de la orientación y la posición pueden ser resueltas independiente y secuencialmente.

Para resumir, se puede utilizar como descriptor de posición la matriz $\mathbf{A}(u, y) \in \mathbb{R}^{k_1 \times k_2}$ y como descriptor de orientación la matriz de argumentos $\Phi(u, y) \in \mathbb{R}^{k_3 \times k_4}$. Durante los experimentos diferentes tamaños son evaluados, para ello las variables k_1, k_2, k_3, k_4 toman diferentes valores. De este modo se puede estudiar por separado la influencia de estos parámetros en la precisión y el coste computacional del proceso de localización.

3.2.2. Descriptores basados en el Histograma de la Orientación de los Gradientes

El Histograma de la Orientación de los Gradientes (HOG) es un tipo de descriptores de características locales típicamente utilizados en visión por computador y en procesamiento de imágenes para realizar la tarea de detección de objetos. El descriptor HOG fue inicialmente descrito por Dalal y Triggs [46], quienes lo usaron para detectar viandantes en secuencias de imágenes. Posteriormente, diferentes investigadores presentaron mejoras de esta versión reduciendo su tiempo computacional y su precisión en la tarea [190]. Hofmeister et al. [63] hicieron uso de HOG para solucionar la tarea de localización de robots móviles mediante imágenes con resolución baja en entornos visualmente simples y cuando la orientación del robot era prácticamente invariable. En [64], los mismos autores presentan una comparativa del método HOG con otro tipo de descriptores, aplicando su uso a pequeños robots en entornos reducidos y controlados, esta evaluación finaliza con resultados similares. Además, Aslan et al. estudian la habilidad del descriptor para realizar tareas de seguimiento cuando hay oclusiones en la imagen [13]. Finalmente, Neumann et al. usan HOG para tareas de detección y localización de coches autónomos [119].

Como se ha mencionado, originalmente el descriptor HOG se definió para describir áreas locales de la escena. En este trabajo se redefine este descriptor para conseguir uno de apariencia global invariable a rotaciones. Esta conversión se realiza utilizando un exhaustivo conjunto de celdas que cubre toda la imagen y permite describirla globalmente. La primera versión de este tipo de descriptor se presenta en [127], donde una

versión global de HOG se utiliza para llevar a cabo una tarea de creación de mapas y una localización Monte Carlo en entornos relativamente extensos. Cuando esta versión de HOG se utiliza para describir imágenes panorámicas, el descriptor permite estimar la posición y orientación del robot.

Siguiendo la filosofía del descriptor HOG, su uso como descriptor de apariencia global consiste esencialmente en calcular el gradiente de la imagen, obteniendo el módulo y la orientación de cada pixel. Si D_x y D_y representan las derivadas de la imagen respecto a los ejes x y y, es posible calcular la magnitud y la orientación de los gradientes como:

$$|G| = \sqrt{D_x^2 + D_y^2} \tag{3.1}$$

$$\theta = \arctan \frac{D_y}{D_x} \tag{3.2}$$

Por tanto, de la imagen inicial $f(x, y) \in \mathbb{R}^{N_1 \times N_2}$ es posible obtener una matriz de magnitudes ($\mathbf{M}(x, y)$) y otra de orientaciones ($\Theta(x, y)$) de los gradientes. En este momento la matriz de argumentos $\Theta(x, y)$ se divide en un conjunto de celdas y, a partir de ellas, se construyen los descriptores. Por un lado, para construir el descriptor de posición se utiliza un conjunto de k_5 celdas horizontales, cuyo ancho es N_2, sin superposición entre celdas y que cubren toda la imagen. Para cada celda se construye un histograma de sus orientaciones compilando la información en b_1 secciones (bins). Durante el proceso, cada pixel de $\Theta(x, y)$ se pesa con el valor de magnitud ofrecido en ese pixel en la matriz de magnitudes $\mathbf{M}(x, y)$. Al final del proceso, el conjunto de histogramas son recopilados en un vector que actuará como descriptor de posición $\vec{h}_1 \in \mathbb{R}^{k_5 \cdot b_1 \times 1}$. El proceso de construcción se puede comprobar en la figura 3.1. Por otro lado, el descriptor de orientación se construye siguiendo los mismos pasos, pero considerando celdas verticales y superpuestas, con una altura igual a N_1 pixeles, un ancho de l_1 y una distancia entre celdas consecutivas de d_1. El número total de celdas verticales se obtiene como $k_6 = N_2/d_1$. Después de compilar el histograma de cada celda en b_2 bins y concatenar estos histogramas, el resultado es un descriptor de orientación $\vec{h}_2 \in \mathbb{R}^{k_6 \cdot b_2 \times 1}$. La figura 3.2 muestra la composición de las celdas para construir los descriptores.

El descriptor $\vec{h}_1$ es invariante a la rotación del robot en el plano del suelo y se puede utilizar como descriptor visual de posición. Por otro lado, la información que contiene $\vec{h}_2$ permite estimar la orientación del robot respecto a otra imagen de referencia.

3.2.3. Descriptores basados en Gist

El descriptor basado en *gist* trata de imitar la habilidad de percepción humana para reconocer inmediatamente una escena por la identificación de regiones específicas que resaltan respecto de sus vecinos. Este concepto fue introducido por primera vez por

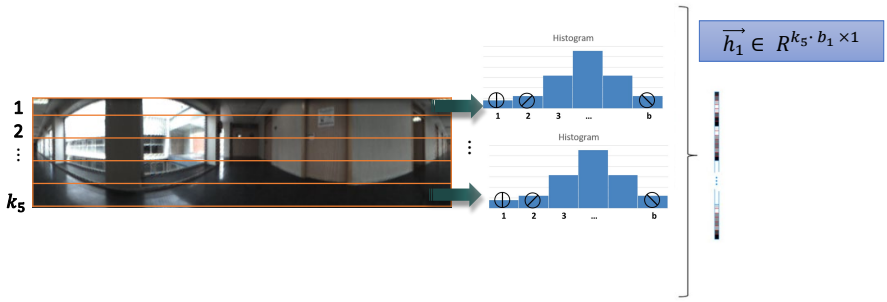


Figura 3.1: Proceso para construir el descriptor de posición HOG con celdas horizontales.

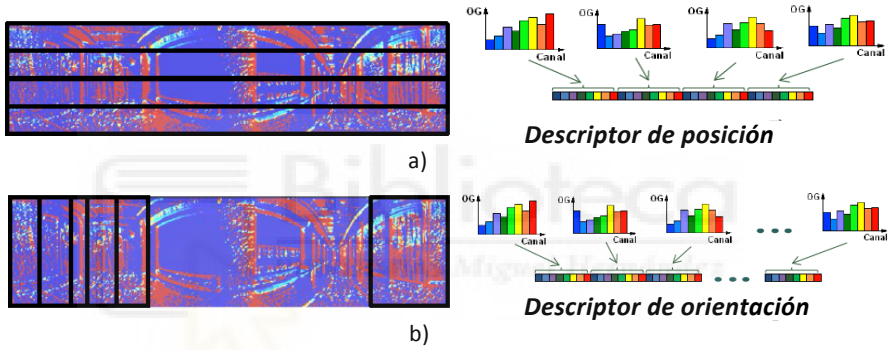


Figura 3.2: Enfoque para construir un descriptor de apariencia global de una imagen panorámica: (**a**) con celdas horizontales (**b**) con celdas verticales superpuestas.

Oliva y Torralba [123, 172] con la idea de crear un descriptor global unidimensional. Más recientemente trabajos relativos a este descriptor hacen uso del concepto de *prominencia* junto al de *gist*. Siagian et al. [155] intentan establecer sinergias entre ambos conceptos obteniendo un único descriptor con las mismas prestaciones pero un coste computacional relativamente reducido. Este descriptor ha sido utilizado habitualmente para tareas de clasificación, mientras que su uso en tareas de robótica móvil es escaso. Alguna aplicación relacionada puede ser encontrada en [38], donde se presenta una tarea de localización y navegación haciendo uso de los conceptos de *gist* y *prominence*; en [115], donde se construyen descriptores *gist* calculados sobre porciones específicas de la imagen panorámica para solucionar una tarea de localización en áreas urbanas; y en [95], donde estos descriptores junto con la reducción PCA (Principal Components Analysis) se usan para resolver un problema de SLAM (Simultaneous Localization and Mapping). Adicionalmente, Su et al. [163] usan *gist* combinado con descriptores de apariencia local para buscar coincidencias entre frames de imágenes. Finalmente, Hays [25] experimenta con el descriptor *gist* para estimar información geográfica de la escena.

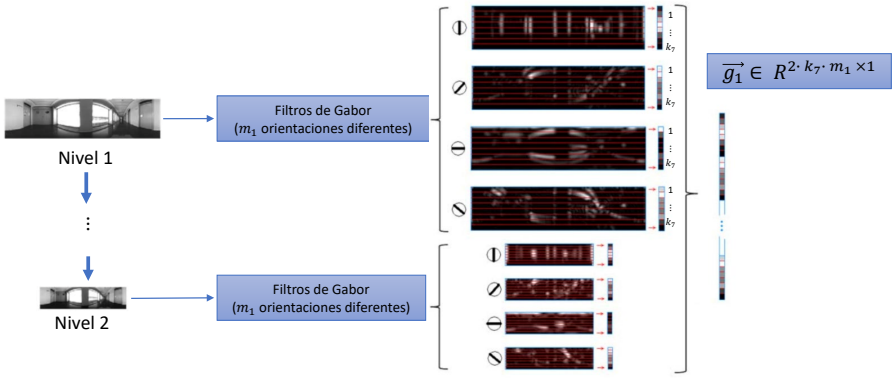


Figura 3.3: Proceso para construir el descriptor de posición *gist* con celdas horizontales.

El método de descripción utilizando en este análisis está basado en los trabajos de Siagian et al. [155] y está descrito profundamente en [127]. Esta versión se construye con información de intensidad obtenida tras aplicar diversos filtros de Gabor con diferentes orientaciones a la imagen en diferentes niveles de resolución. Primero, se consideran dos versiones de la imagen panorámica original: la original y una nueva versión con una resolución menor de $0{,}5 \cdot N_1 \times 0{,}5 \cdot N_2$. Después de esto, ambas imágenes se filtran con el conjunto de m_1 filtros de Gabor cuyas orientaciones se distribuyen entre 0 y 180 grados. Finalmente, se reduce la información agrupando los pixeles en bloques o celdas y calculando la intensidad media de los pixeles que caigan en esta región y concatenando los valores obtenidos. Como con HOG, existe la versión de utilizar celdas horizontales no superpuestas y crear el descriptor de posición $\vec{g}_1 \in \mathbb{R}^{2 \cdot k_7 \cdot m_1 \times 1}$, este descriptor es invariante a la orientación del robot respecto al plano del suelo. Por otro lado también es posible crear el descriptor utilizando bloques verticales y superpuestos y definir de esta manera el descriptor de orientación $\vec{g}_2 \in \mathbb{R}^{2 \cdot k_8 \cdot m_2 \times 1}$. Esta filosofía se puede observar en la figura 3.2. La figura 3.3 muestra el proceso de construcción del descriptor de posición *gist* con celdas horizontales.

3.2.4. Descriptores basados en Wi-SURF

El descriptor 'Speeded-Up Robust Features' (SURF) [18] se considera uno de los descriptores de apariencia local más importantes y ha sido utilizado en diversos trabajos para resolver tareas de localización con considerable eficiencia, como [7] y [116]. En este estudio se ha adaptado su uso para construir descriptores de apariencia global. Para ello, se propone seguir la adaptación de trabajos previos [1], en el cual se extrae un único descriptor de apariencia global de la imagen calculando en diferentes ventanas el descriptor SURF. En este estudio presentamos este descriptor como Wi-SURF (Whole Image SURF).

Nuestra idea del descriptor Wi-SURF se ha usado parcialmente en trabajos anteriores para tareas de localización topométrica [14] y para reconocimiento de lugares

[10]. Estos trabajos proponen obtener un único descriptor SURF $d \in \mathbb{R}^{64}$ que contiene información de toda la imagen. Este descriptor puede ser útil para reconocimiento de estancias, pero no contiene información para estimar la orientación. Asimismo, un único descriptor de 64 valores no resulta lo suficientemente profundo como para captar todas las características visuales necesarias. Por estas razones, se propone dividir la imagen panorámica en un conjunto de ventanas cuadradas distribuidas uniformemente, con algo de superposición entre ellas. En cada ventana, se calcula el descriptor SURF $d \in \mathbb{R}^{64}$ y se van concatenando los diferentes descriptores, obteniendo así el descriptor de apariencia global Wi-SURF. Este descriptor permite su uso, no solo como descriptor de posición, sino que también se puede utilizar como descriptor de orientación, como se detalla en la sección 3.3.4. Las ventanas cuadradas están distribuidas de manera uniforme siguiendo los siguientes parámetros: k_9 es el número de celdas horizontales en las que la imagen panorámica se ha dividido y sp_1 es el espacio horizontal entre ventanas consecutivas. El número de ventanas por celdas dependerá del ancho de la imagen (en nuestro caso 512 columnas), por tanto, en nuestros experimentos cada celda horizontal se divide en $w_1 = \frac{512}{sp_1}$ ventanas. El ancho de cada ventana cuadrada debe ser igual a la altura de la celda. Después del proceso de concatenación de vectores SURF, se obtiene el descriptor WS $\vec{ws} \in \mathbb{R}^{k_9 \cdot w_1 \cdot 64 \times 1}$. Este descriptor es válido tanto para estimar la posición como la orientación.

3.2.5. Descriptores basados en BRIEF-Gist

El descriptor BRIEF-gist es un descriptor de apariencia global basado en el descriptor local 'Binary Robust Independent Elementary Features' (BRIEF). BRIEF se presentó en [27] y se ha utilizado en diferentes aplicaciones de robótica [187, 2]. Su principal característica es su bajo coste computacional cuando se utiliza como descriptor de apariencia local. Basándose en este descriptor, se presenta el descriptor de apariencia global BRIEF-gist [166]. Esta adaptación se ha utilizado para reconocimiento de escenarios y tareas de detección de cierre de bucle [10]. En la presente tesis doctoral, el descriptor de apariencia global se ha adaptado para ser usado en imágenes panorámicas y de este modo obtener un descriptor válido para cálculos de posición y orientación relativa en tareas de localización.

Para implementar el descriptor BRIEF-gist utilizando en esta tesis, se divide la imagen en $k_{10} \times w_2$ ventanas cuadradas no solapadas y del mismo tamaño. A continuación, siguiendo la filosofía del descriptor BRIEF, se calcula la intensidad media de cada ventana y se definen pares de píxeles ordenados entre ventanas contiguas. A continuación, se comparan los valores relativos de cada par de píxeles adyacentes. Si la diferencia de intensidad relativa entre una ventana y su pretérita es positiva se añade un 1 al vector descriptor, por su parte si la diferencia es negativa se añade un 0. Como resultado se obtiene un vector booleano. Después de este proceso se obtiene el descriptor BRIEF-gist $\vec{bg} \in \mathbb{R}^{k_{10} \cdot w_2 \times 1}$, que puede ser utilizado como descriptor de posición y de orientación como se puede comprobar en la sección 3.3.5.

3.2.6. Descriptores basados en Radon Transform

La transformada Radon es propuesta en [137]. Inicialmente se utilizó en aplicaciones de visión por computador como un descriptor de formas geométricas, como en [62, 60]. Más recientemente, la transformada de Radon (RT) se ha adaptado para describir de manera global imágenes omnidireccionales. Este trabajo se testeó en [20], donde un descriptor basado en RT se usó para solucionar el problema de detección de revisita de entornos conocidos, y en [21], donde los descriptores se usaron para estimar la altura relativa de las imágenes. La principal ventaja de estos descriptores es que pueden calcularse con la imagen omnidireccional capturada por el sistema de visión (sin necesidad de transformarla a panorámica).

Matemáticamente, la transformada Radon consiste en describir una función en términos de las proyecciones de sus integrales de línea.

Tras aplicar la transformada de Radon, toda la imagen se transforma en una función $r_{im}(\Phi, d)$, obtenida después de integrar la función original a través de diferentes grupos de líneas paralelas con una distancia al origen d y diferente orientación Φ. El tamaño del nuevo descriptor es $r_{im} \in \mathbb{R}^{M_x \times M_y}$, donde M_x es el número de orientaciones en las que se calcula la integral ($\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_{M_x}\}$) y M_y es el número de líneas paralelas.

Cuando la transformada de Radon se aplica sobre imágenes omnidireccionales, es interesante estudiar su simetría y el hecho de que el descriptor gire horizontalmente cuando el robot rota [19]. Esta última característica permite tener la posibilidad de estimar la orientación del robot. Esta propiedad se puede ver en la figura 3.4, donde se presentan cuatro imágenes; tres de ellas han sido capturadas desde la misma posición pero con distinta orientación y la otra ha sido capturada desde una posición diferente. La figura muestra claramente el efecto de la orientación en la transformada de Radon y como varía la transformada cuando la imagen pertenece a una habitación diferente. Si el robot rota ($\Delta\theta$) grados, el nuevo descriptor presenta la misma información que la imagen original pero trasladada s columnas, $s = (\Delta\theta) \cdot (M\_x)/360$. Gracias a esta propiedad el descriptor calculado con la transformada de Radon permite estimar la orientación del robot y el descriptor puede utilizarse como descriptor de posición y de orientación.

Para resumir, tras aplicar la transformada Radon a la imagen omnidireccional de tamaño $N_x \times N_x$, se obtiene una matriz $r \in \mathbb{R}^{\frac{360}{p_1} \times 0{,}5 \cdot N_x}$, donde p_1 es el ángulo (grados) entre los conjuntos de líneas paralelas consecutivas en las cuales se calculará las integrales de línea. En los experimentos, estas matrices pueden usarse de diferente manera, dando pie a dos submétodos (el método Radon-Fourier y el método Radon-POC). Estos métodos trabajan el descriptor de manera diferente y se evaluará ambos para determinar la robustez de ambos. Ambos métodos y sus parámetros están descritos en la sección 3.3.6.

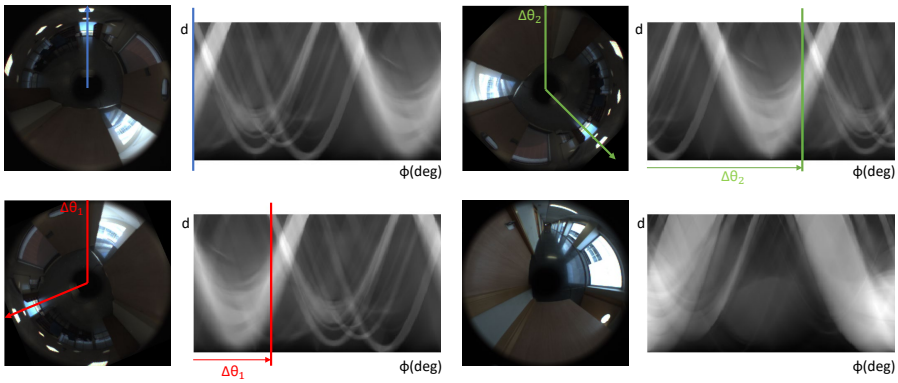


Figura 3.4: Propiedad de giro en la Transformada Radon.

3.3. Resolución del problema de localización absoluto

Para poder realizar la tarea de localización, en este trabajo, se asume que el modelo del entorno está disponible previamente, es decir, que en memoria el robot dispone de unos puntos de localización con su imagen y descriptor correspondiente. Para construir este modelo, el robot se desplaza por el entorno inicialmente desconocido (esta tarea puede realizarse teleoperada o utilizando algún algoritmo de exploración [72, 93]) y captura un conjunto de imágenes omnidireccionales desde n puntos de vista, definidos por sus poses $\vec{p}_j = (x_j, y_j, \theta_j), j = 1, \ldots, n$, que cubren todo el entorno por el que posteriormente se prevé localizar. Para que el modelo sea funcional, es necesario que este contega la suficiente infomación de las imágenes capturadas para que sea posible estimar la posición y orientación del robot cuando una nueva imagen sea capturada (teniendo en cuenta que el robot puede tener posición y orientación diferente a las capturadas en el modelo) [131]. Para ello, el modelo $\mathcal{M}$ está compuesto por los descriptores de las imágenes y las poses del robot en el punto donde se tomó la imagen. Esta información se almacena conjuntamente: $\mathcal{M} = \{(\mathcal{D}_1, \vec{p}_1), (\mathcal{D}_2, \vec{p}_2), \ldots (\mathcal{D}_n, \vec{p}_n)\}$, y donde, en general, la descripción de la imagen consiste en un descriptor de posición y otro de orientación $\mathcal{D}_j = \{\vec{d}_{1j}, \vec{d}_{2j}\}$ (en el caso de Wi-SURF y BRIEF-gist el mismo vector se usa como descriptor de posición y de orientación, en estos casos $\mathcal{D}_j = \{\vec{d}_{1j}\}$).

Una vez el modelo está creado, el problema de localización consiste en estimar la pose actual del robot. El problema se asume como una tarea de localización absoluta, ya que no se considera información de la posición previa del robot y solo se utiliza información visual. El robot captura una imagen nueva en el instante t, desde una pose desconocida (f_t, *test image*). Entonces, se calcula el descriptor de la imagen $\mathcal{D}_t$, este se compara con el conjunto de imágenes guardadas en el modelo. Tras esta comparación, se estima la posición y la orientación del robot en el instante t. En las siguientes subsecciones se detalla el proceso dependiendo del método de descripción.

3.3.1. Localización utilizando descriptores basados en la Transformada Discreta de Fourier

Cuando una nueva imagen de test llega, se calculan sus matrices $\mathbf{A}_t$ y Φ_t. La matriz $\mathbf{A}_t$ compone el descriptor de posición y este es invariante a rotaciones en el plano del suelo, por ello, $\mathbf{A}_t$ se utiliza para estimar la posición del robot comparándolo con los distintos descriptores almacenados $\mathbf{A}_j$, $j = 1, \ldots, n$ y seleccionando los k vecinos más cercanos. La posición del vecino más cercano (x_i, y_i) (i es el índice del vecino más cercano) puede considerarse como la estimación de la posición del robot en el instante t. Una vez que la posición del robot ha sido estimada, la matriz de argumentos de la imagen test Φ_t, y la matriz de argumentos del vecino más cercano Φ_i, se usan para estimar la orientación relativa del robot, utilizando la propiedad de desplazamiento de la Transformada Discreta de Fourier. El objetivo es estimar la orientación relativa θ_{ti} en el instante t con respecto de la orientación que tenía el robot cuando el vecino más cercano fue capturado, $\theta_{ti} = \theta_t - \theta_i$. El proceso de estimación de la rotación sigue los siguientes puntos:

1. La imagen test es rotada artificialmente. El cálculo de los descriptores basados en la Transformada Discreta de Fourier permite agregar rotaciones artificiales a las imágenes.

   La propiedad de desplazamiento de la Transformada de Fourier permite determinar la rotación producida entre imágenes capturadas en el mismo punto, pero con distinta orientación. El paso entre cada par de rotaciones artificiales consecutivas es $\Delta\phi$. Esto es equivalente a realizar una rotación entre columnas de la imagen panorámica con una magnitud de d píxeles, donde $\Delta\phi = d \cdot 2\pi/N_2$. En los experimentos consideramos $d = \{1, 2, \ldots, N_2 - 1\}$. Esto significa que el paso angular entre rotaciones artificiales consecutivas será de $\Delta\phi = 2\pi/N_2$. Esta será la resolución de nuestro método.

2. Después de este proceso, un conjunto de $n_{rot} = 2\pi/\Delta\phi$ matrices de argumentos son obtenidas para la imagen del instante t.

$$\{\Phi_0, \Phi_1, \ldots, \Phi_{n_{rot}}\}_t = \{\Phi_\alpha\}_t, \alpha = 0, \ldots, n_{rot} \tag{3.3}$$

3. Se calcula el producto Hadamard de la matriz Φ_t y cada matriz Φ_α. Se obtiene la suma de las componentes de cada matriz resultante, y el resultado es un vector de datos:

$$\{m_0, m_1, \ldots, m_{n_{rot}}\}_t = \{m_\alpha\}_t, \alpha = 0, \ldots, n_{rot} \tag{3.4}$$

4. La rotación relativa estimada es α, cuyo valor corresponde con el coeficiente de m_α con mayor valor.

$$\alpha = arg\,max_\alpha\{m_\alpha\} \tag{3.5}$$

$$\theta_{ti} = \frac{2\pi\alpha}{n_{rot}} \tag{3.6}$$

donde θ_{ti} es la orientación relativa entre la imagen im_t y el vecino más cercano im_i. De este modo, la orientación absoluta del robot en el instante t se puede calcular como:

$$\theta_t = \theta_i + \theta_{ti} \tag{3.7}$$

En esta ecuación, θ_i es la orientación que el robot tenía cuando la imagen im_i del mapa fue capturada con respecto a una referencia global.

En los experimentos, los parámetros a optimizar de la Firma de Fourier son el tamaño de la matriz de módulos (k_1 y k_2) y el tamaño de la matriz de argumentos (k_3 y k_4). La finalidad es llegar a un balance entre precisión en la estimación de la posición y orientación y el coste computacional del algoritmo.

3.3.2. Localización utilizando descriptores basados en el Histograma de la Orientación de los Gradientes

Una vez obtenida la imagen im_t, se calcula los descriptores $\vec{h}_{1t}$ y $\vec{h}_{2t}$. Primero, $\vec{h}_{1t}$ se compara con los descriptores almacenados $\vec{h}_{1j}$, $j = 1, \ldots, n$ y se calcula el vecino más cercano. A continuación se extrae la posición (x_i, y_i) del que se ha determinado que es el vecino más cercano i y esta posición será la que determine la posición del robot móvil en el instante t.

A continuación, la orientación del robot se calcula comparando el vector $\vec{h}_{2t}$ con el descriptor de orientación del vecino más cercano $\vec{h}_{2i}$. Con este espíritu se calcula un conjunto de rotaciones artificiales a partir del vector $\vec{h}_{2t}$. Después, se realiza el producto escalar entre los vectores rotados y el descriptor de orientación del vecino más cercano $\vec{h}_{2i}$. El producto con un mayor valor del producto escalar será el que determine la rotación de robot. Para simular artificialmente la rotación del robot en el instante t es necesario rotar el vector $\vec{h}_{2t}$, para ello se realiza un desplazamiento circular de las columnas del vector con un paso múltiplo de b_2 (b_2 es el número de bins por histograma en el descriptor de orientación). Una desplazamiento circular del vector de b_2 posiciones equivale a una rotación del robot de valor $\Delta\phi = 2\pi d_1/N_2$ radianes (este será la resolución angular del método), donde d_1 es la distancia entre dos celdas verticales.

Finalmente, se estima la orientación relativa θ_{ti}. Este valor es el ángulo entre el descriptor de orientación base $\vec{h}_{2i}$ y la versión rotada del vector $\vec{h}_{2t}$ que dió mayor producto escalar.

3.3.3. Localización utilizando descriptores basados en Gist

Los procesos de estimar la posición y orientación son idénticos a los presentados en el caso de HOG. Una vez se captura la imagen de test im_t, se calculan los descriptores $\vec{g}_{1t}$ and $\vec{g}_{2t}$. Primero, se compara el descriptor de posición $\vec{g}_{1t}$ con los descriptores gist almacenados $\vec{g}_{1j}$, $j = 1, \ldots, n$ y se calculan los vecinos más cercanos. Una vez detectados, se selecciona el vecino más cercano i y su posición $(x, y)_i$ será considerada la posición del robot en el instante t. Después de esto, se trata de estimar la orientación relativa entre la pose en el instante i y en el instante t. Para ello se utilizan los descriptores de orientación y se compara el descriptor $\vec{g}_{2i}$ con una serie de vectores resultantes de rotar artificialmente el descriptor de orientación actual $\vec{g}_{2t}$. Este descriptor se rota artificialmente desplazando circularmente su vector con un paso múltiplo de m_2 (m_2 es el número de componentes de cada bloque vertical). Cada giro equivale a una rotación de $\Delta\phi = 2\pi d_2/N_2$ radianes (resolución angular del método), donde d_2 es la distancia entre dos bloques verticales consecutivos.

El valor resultante θ_{ti}, es el ángulo correspondiente entre la versión rotada de $\vec{g}_{2t}$ que presenta el mayor producto escalar con el vector $\vec{g}_{2i}$.

3.3.4. Localización utilizando descriptores basados en Wi-SURF

Una vez obtenida la imagen im_t, se calcula el descriptor $\vec{ws}_t$. Como se ha comentado en la sección anterior, este descriptor es único tanto para posición como para orientación. Primero, el descriptor se compara con los datos almacenados del modelo creado $\vec{ws}_j, j = 1, \ldots, n$, y se giran artificialmente todos los descriptores del modelo de forma que tengan la misma orientación relativa que el descriptor de la imagen de test. Para estimar la orientación relativa y conseguir que todos los descriptores del modelo tengan la misma orientación relativa que $\vec{ws}_t$, se añaden diferentes rotaciones artificiales al descriptor $\vec{ws}_t$ y se calcula la distancia entre cada uno de los descriptores resultantes y $\vec{ws}_j$. Se selecciona la rotación que da lugar a la distancia más pequeña y su correspondiente θ_{tj} es el ángulo relativo entre ambos descriptores. Para simular las rotaciones artificiales de $\vec{ws}_t$, se aplica la técnica del desplazamiento circular. Cada giro debe ser múltiplo de 64 (cada descriptor SURF contiene 64 componentes) y de w_1 (número de ventanas). Un giro de 64 componentes del descriptor es equivalente a una rotación del robot de $\Delta\phi = 2 \cdot \pi \cdot sp_1/N_2$ radianes (y por tanto, esta es la resolución angular del método). Una vez la orientación relativa entre la imagen de test im_t y la imagen del modelo im_j ha sido calculada, los descriptores son orientados hacia la misma posición.

Una vez que todos los descriptores están orientados del mismo modo, se calculan los vecinos más cercanos a $\vec{ws}_t$ se calculan con la distancia entre vectores. La posición (x_i, y_i) del vecino más cercano i es la estimación de la posición del robot en el instante t. La orientación entre ambas imágenes se había calculado previamente y corresponde con el ángulo θ_{ti}, que es el giro necesario para orientar los descriptores del mismo modo.

3.3.5. Localización utilizando descriptores basados en BRIEF-Gist

Tras recibir la nueva imagen im_t, el sistema calcula su descriptor $\vec{bg}_t$. Como en el caso del descriptor Wi-SURF, un único descriptor realiza la función de descriptor de posición y de orientación. Primero, se estima la diferencia de orientación entre el descriptor de la imagen test y los descriptores BRIEF-Gist del modelo $\vec{bg}_j, j = 1, \ldots, n$. Para calcular esta orientación relativa entre descriptores, se aplican sucesivas rotaciones artificiales al descriptor $\vec{bg}_t$. Por cada imagen del modelo se calcula la distancia frente a todas las rotaciones del vector $\vec{bg}_t$, y la comparación con menor distancia entre descriptores es elegida para calcular el cambio de orientación. Para simular las rotaciones artificiales de $\vec{bg}_t$, se utiliza el desplazamiento circular del descriptor, este desplazamiento debe ser múltiplo de w_2 (número de ventanas en cada celda). La resolución del método depende de este valor, cada desplazamiento de w_2 equivale a una rotación del robot de $\Delta\phi = 2 \cdot \pi / w_2$ radianes.

Una vez estimada la orientación relativa con cada descriptor del modelo $\vec{bg}_j$ y girados los vectores para que tengan la misma orientación, es posible calcular la distancia entre descriptores. Se seleccionan los vecinos más cercanos a $\vec{bg}_t$ calculando la distancia a los descriptores del modelo cuando tienen todos la misma orientación. La posición (x_i, y_i) del vecino más cercano (i) será tomada como estimación del robot en el instante (t). La orientación relativa entre la imagen test y el vecino más cercano ya ha sido calculada en una primera fase. θ_{ti} es el ángulo necesario para que $\vec{bg}_t$ y el descriptor $\vec{bg}_j$ tengan la misma orientación.

3.3.6. Localización utilizando descriptores basados en la Transformada Radon

Para una exhaustiva evaluación, este descriptor se va a tratar de dos métodos distintos, estudiando dos maneras de calcular posición y orientación del robot.

3.3.6.1. Método Radon–Fourier

Tras calcular la transformada de Radon, el sistema obtiene una matriz $r \in \mathbb{R}^{\frac{360}{p_1} \times 0{,}5 \cdot N_x}$ de la imagen de test im_t. En este primer método, la Firma de Fourier (FS) se aplica a la matriz obtenida de la transformada Radon. Como resultado de esta segunda transformación se obtiene una matriz de módulos $\mathbf{A_{RT}}_j \in \mathbb{R}^{\frac{360}{p_1} \times k_{11}}$ y otra de argumentos $\Phi_{RT\,j} \in \mathbb{R}^{\frac{360}{p_1} \times k_{12}}$. A partir de este punto, las matrices se tratan igual que en el caso de los descriptores basados en la Transformada Discreta de Fourier. $\mathbf{A_{RT}}_j$ se utiliza como descriptor de posición y $\Phi_{RT\,j}$ como descriptor de orientación. En este caso, el parámetro k_{11} es el número de columnas retenidas del descriptor de posición $\mathbf{A_{RT}}_j$, mientras que k_{12} es el número de columnas retenidas del descriptor de orientación $\Phi_{RT\,j}$. El método para estimar la posición y la orientación del robot en el instante (t) sigue el mismo proceso que el método usando descriptores basados en la Transformada de Fourier, presentado en la sección 3.3.1.

3.3.6.2. Método Radon–POC

Este método usa directamente la matriz que se obtiene tras aplicar la transformada de Radon, es decir, de la imagen test im_t se obtiene la matriz $r \in \mathbb{R}^{\frac{360}{p_1} \times 0{,}5 \cdot N_x}$, que funciona como descriptor r_{poc_j}. Para comparar dos descriptores se usa la técnica Phase Only Correlation (POC). Esta operación tiene como resultado un coeficiente de correlación que permite tanto estimar la similitud entre dos matrices como el giro relativo entre ambas.

El descriptor de la imagen test r_{poc_j} se compara con todos los descriptores del modelo $r_{\vec{poc}_j}, j = 1, \ldots, n$. Cada comparación da un coeficiente de similitud entre los descriptores. La imagen que supone la comparación POC con el coeficiente de similitud más alto es seleccionada como la más cercana. La posición de esta imagen i se estimará como posición del robot en el instante t. Para estimar la orientación relativa del robot se hace uso de la salida de POC. La posición del máximo valor de coeficiente corresponde con el giro necesario entre las matrices. La resolución del sistema $\Delta\phi$ corresponde con el parámetro p_1.

Para resumir, el cuadro 3.1 muestra los parámetros con influencia en los descriptores y cuyos valores se evalúan en este estudio. Después de esto, el cuadro 3.2 da detalles del tamaño y forma de los descriptores dependiendo del método empleado.

Cuadro 3.1: Parámetros cuya influencia en el proceso de localización es estudiada.

Descr.	Parámetros
FS	$k_1 \Rightarrow$ filas en el descriptor de posición $\mathbf{A}_j$ $k_2 \Rightarrow$ columnas en el descriptor de posición $\mathbf{A}_j$ $k_3 \Rightarrow$ filas en el descriptor de orientación Φ_j $k_4 \Rightarrow$ columnas en el descriptor de orientación Φ_j
HOG	$b_1 \Rightarrow$ bins por histograma en el descriptor de posición $\vec{h}_{1j}$ $k_5 \Rightarrow$ celdas horizontales en el descriptor de posición $\vec{h}_{1j}$ $b_2 \Rightarrow$ bins por histograma en el descriptor de orientación $\vec{h}_{2j}$ $l_1 \Rightarrow$ ancho de las celdas en el descriptor de orientación $\vec{h}_{2j}$ $d_1 \Rightarrow$ distancia entre celdas , descriptor de orientación $\vec{h}_{2j}$ $k_6 = \frac{N_2}{d_1} \Rightarrow$ celdas en el descriptor de orientación $\vec{h}_{2j}$
Gist	$m_1 \Rightarrow$ orientaciones (filtro de Gabor), descriptor de posición $\vec{g}_{1j}$ $k_7 \Rightarrow$ bloques horizontales en el descriptor de posición $\vec{g}_{1j}$ $m_2 \Rightarrow$ orientaciones (filtro de Gabor), descriptor de orientación $\vec{g}_{2j}$ $l_2 \Rightarrow$ ancho de los bloques, descriptor de orientación $\vec{g}_{2j}$ $d_2 \Rightarrow$ distancia entre bloques, descriptor de orientación $\vec{g}_2$ $k_8 = \frac{N_2}{d_2} \Rightarrow$ bloques, descriptor de orientación $\vec{g}_{2j}$
WS	$w_1 \Rightarrow$ ventanas por celda en el descriptor $\vec{ws}_j$ $k_9 \Rightarrow$ bloques horizontales en el descriptor $\vec{ws}_j$ $sp_1 \Rightarrow$ espacio horizontal entre ventanas, descriptor $\vec{ws}_j$
BG	$w_2 \Rightarrow$ ventanas por celda en el descriptor $\vec{bg}_j$ $k_{10} \Rightarrow$ bloques horizontales en el descriptor $\vec{bg}_j$
RT	$p_1 \Rightarrow$ grados entre lineas en las que se calcula Radon en r $k_{11} \Rightarrow$ columnas en el descriptor de posición $\mathbf{A_{RT}}_j$ $k_{12} \Rightarrow$ columnas en el descriptor de orientación $\Phi_{RT\,j}$ $N_x \Rightarrow$ el tamaño de la imagen omnidireccional es $N_x \times N_x$

Cuadro 3.2: Tamaño de descriptor de apariencia global de cada imagen para las tareas de localización y estimación de orientación.

Descriptor	Localización	Orientación
FS	$\mathbf{A}_j \in \mathbb{R}^{k_1 \times k_2}$	$\Phi_j \in \mathbb{R}^{k_3 \times k_4}$
HOG	$\vec{h}_{1j} \in \mathbb{R}^{k_5 \cdot b_1 \times 1}$	$\vec{h}_{2j} \in \mathbb{R}^{k_6 \cdot b_2 \times 1}$
Gist	$\vec{g}_{1j} \in \mathbb{R}^{2 \cdot k_7 \cdot m_1 \times 1}$	$\vec{g}_{2j} \in \mathbb{R}^{k_8 \cdot m_2 \times 1}$
WS	$\vec{ws}_j \in \mathbb{R}^{k_9 \cdot w_1 \cdot 64 \times 1}$	
BG	$\vec{bg}_j \in \mathbb{R}^{k_{10} \cdot w_2 \times 1}$	
RT–F	$\mathbf{A_{RT}}_j \in \mathbb{R}^{\frac{360}{p_1} \times k_{11}}$	$\Phi_{RT\,j} \in \mathbb{R}^{\frac{360}{p_1} \times k_{12}}$
RT–POC	$r_{poc_j} \in \mathbb{R}^{\frac{360}{p_1} \times 0{,}5 \cdot N_x}$	

3.4. Configuración de los experimentos

Este apartado describe la configuración de los experimentos. Primero, se muestra la información relativa al conjunto de imágenes utilizado para la experimentación. A continuación, se presentan distintos fenómenos (ruido y oclusiones) con los que se va a experimentar los descriptores, estos fenómenos ayudarán a concretar la robustez de los algoritmos.

3.4.1. Conjunto de imágenes

Todos los experimentos se ejecutan utilizando una base de datos captada por el grupo de investigación ARVC de la Universidad Miguel Hernández de Elche [12]. Para ello, se utiliza un sistema de visión catadióptrico. Este sistema está compuesto por una cámara *Imaging Source DFK 21BF04* alineada con el eje de un espejo hiperbólico *Eizoh Wide 70* (figura 3.5). Este sistema es capaz de capturar imágenes omnidireccionales que son preprocesadas para obtener sus proyecciones cilíndricas (imágenes panorámicas) con tamaño $N_1 \times N_2 = 128 \times 512$ píxeles.

Un primer conjunto de imágenes, a las que denominamos *imágenes de entrenamiento*, está compuesto por 872 imágenes panorámicas capturadas en un mapa de rejilla de tamaño 40×40 cm, que cubre toda una planta de uno de los edificios de la Universidad Miguel Hernández (España) y que incluye 6 habitaciones diferentes.

Un segundo conjunto de imágenes, conocidas como *imágenes de test* contiene 1232 imágenes capturadas en todas las habitaciones. Estas imágenes poseen diferentes orientaciones. Para obtener este conjunto de imágenes se capturan 93 imágenes test, 77 de ellas en posiciones intermedias que no corresponden con puntos de la rejilla donde se capturaron imágenes de entrenamiento y 16 en posiciones de la rejilla, pero con una orientación del robot diferente a las imágenes de entrenamiento. Las imágenes de test fueron capturadas en días diferentes y en momentos del día distintos. De esta manera pueden aparecer cambios en la posición de algunos objetos, puertas, personas

Figura 3.5: Cámara *Imaging Source DFK 21BF04* y espejo hiperbólico *Eizoh Wide 70* alineados para la captura de las imágenes omnidireccionales.

transitando por el entorno o pequeños cambios en la iluminación, efectos que simularán una posible variación de la información visual tal y como puede suceder en entornos de trabajo reales y dinámicos. El conjunto de imágenes test será utilizado durante el proceso de localización y estimación de la orientación para evaluar la eficiencia de cada uno de los métodos de descripción global y la influencia de sus parámetros. El entorno utilizado para realizar esta evaluación es muy propenso a sufrir el problema de *aliasing perceptual*. Este fenómeno surge cuando dos imágenes capturadas desde dos puntos diferentes contienen una información visual parecida. Esto puede suceder en entornos de oficinas, despachos, habitaciones de hospital... Los métodos de apariencia global deben ser capaces de combatir este fenómeno ya que es común que pase en entornos de interior.

La figura 3.6 muestra una vista en planta del entorno y los puntos de captura de las imágenes de entrenamiento. Como ejemplo, la figura 3.7 muestra la sala de la habitación 5 en la figura 3.6. En esta figura se puede observar los puntos de captura de las imágenes, en rojo están las imágenes de entrenamiento y de color verde los puntos de captura de las imágenes de test. Además se muestran los cambios en las condiciones de iluminación y de orientación entre las imágenes test y de entrenamiento. Otro ejemplo de espacio aparece en la figura 3.8, donde se representa el pasillo. En este caso se puede detectar el efecto de *aliasing perceptual*. Además, la imagen de test 3 muestra un ejemplo de cambios en el entorno (hay una puerta abierta respecto de la imagen de entrenamiento).

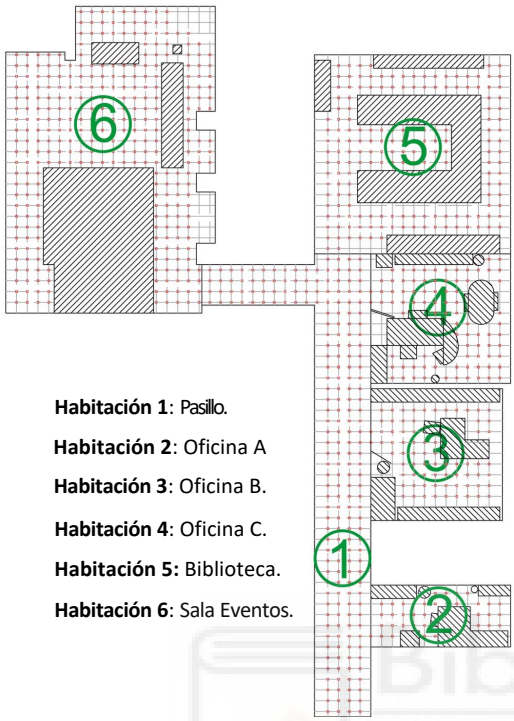


Figura 3.6: Vista en planta de los puntos de captura del conjunto de imágenes de entrenamiento. El tamaño de la rejilla es de 40 × 40 cm.

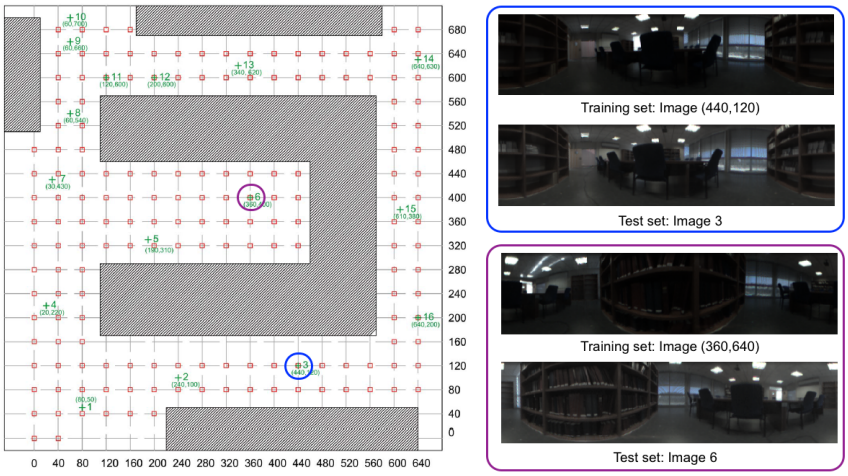


Figura 3.7: Biblioteca. Vista en planta de los puntos de captura de las imágenes de entrenamiento. El tamaño de la rejilla es de 40 × 40 cm.

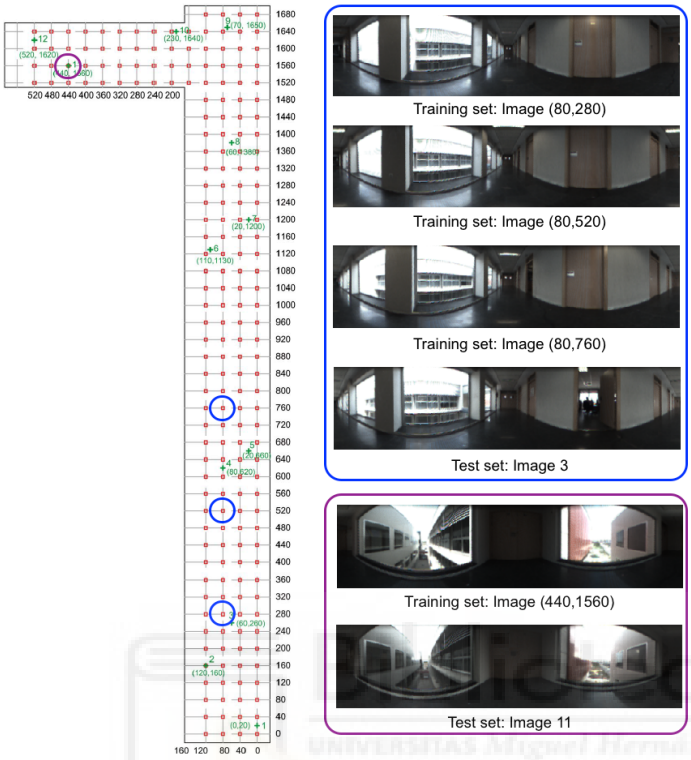


Figura 3.8: Pasillo. Vista en planta de los puntos de captura de las imágenes de entrenamiento. El tamaño de la rejilla es de 40×40 cm.

3.4.2. Efectos de ruido y oclusiones

Las imágenes de test reflejan los efectos reales en la visualización más comunes en un entorno de trabajo: los cambios en las condiciones de iluminación, en la posición y estado de algunos objetos y el aliasing percepual. Adicionalmente, dos fenómenos más se van a tener en cuenta durante la realización de los experimentos, estos son el ruido y las oclusiones.

Primero, para estudiar el efecto del ruido en los sistemas de adquisición, se añade a las imágenes un ruido Gaussiano. Este fenómeno se considera añadiendo a la imagen una distribución Gaussiana de valores con media nula y diferentes niveles de varianza: $\sigma^2 = \{0, 0'0025, 0'05, 0'01, 0'02, 0'05\}$. A lo largo del resto del capítulo se denominará a estos niveles 0, 1, 2, 3, 4 y 5 respectivamente. La figura 3.9a muestra una imagen test con diferentes niveles de ruido añadido. En los casos más extremos, la apariencia visual de las imágenes se altera visiblemente.

Adicionalmente, la presencia de personas, otros robots o incluso objetos en el entorno puede crear una oclusión parcial y temporal de la imagen. Trabajar con imágenes panorámicas puede suponer una ventaja si aparecen oclusiones en la escena. Sin

embargo, estas oclusiones pueden ocultar información importante y características relevantes para el proceso de localización. Para modelar este efecto de oclusiones, se añade artificialmente a la imagen diferentes niveles de oclusión, considerando diferentes barras verticales. Estas barras simulan una oclusión de la imagen de un $\{0, 5, 10, 20, 40\}\,\%$. A lo largo de este capítulo nos referimos a esos niveles de oclusión como 0, 1, 2, 3 y 4 respectivamente. La figura 3.9b muestra una imagen test con los diferentes niveles de oclusión añadidos. En el caso más extremo se ha perdido el $40\,\%$ de la información visual.

(a) (b)

Figura 3.9: Imagen de muestra y efectos incluidos. (**a**) Con diferentes niveles de ruido Gaussiano ($\sigma^2 = \{0, 0'0025, 0'05, 0'01, 0'02, 0'05\}$) y (**b**) diferentes porcentajes de oclusiones ($\{0, 5, 10, 20, 40\}\,\%$).

3.5. Resultados y discusión

En esta sección, se realiza un exhaustivo banco de experimentos para evaluar el uso de los descriptores de apariencia global. La sección incluye un estudio de la influencia de los parámetros en la precisión y coste computacional del proceso de estimación de posición y orientación relativa. Esta sección se divide en cuatro subsecciones. Primero, en la subsección 3.5.1 se evalúa la habilidad de cada descriptor para detectar

el vecino más cercano en el modelo. Este estudio se realiza en condiciones ideales (se consideran los problemas de ruido y oclusiones). A continuación, se resuelve el problema de localización, incluyendo los efectos de ruido y oclusiones (subsección 3.5.2). Además, en la subsección 3.5.3, se resuelve el problema de estimación de la orientación del robot. Finalmente, en la subsección 3.5.4 se extiende el estudio del desempeño de los descriptores para resolver la tarea de localización con un conjunto de imágenes capturadas a lo largo de una trayectoria real.

3.5.1. Problema de localización. Vecino más cercano

Durante el proceso de localización del robot móvil, el primer paso consiste en comparar los descriptores de localización de las imágenes de test con los descriptores de localización de las imágenes de entrenamiento (imágenes del modelo), obteniendo para cada imagen test los k-vecinos más cercanos. Teniendo esto en cuenta, en esta sección se explica el método empleado para calcular los vecinos más cercanos, es decir, para detectar los descriptores del modelo más similares al nuevo descriptor test. A este proceso se le domina problema de localización.

Para obtener los k-vecinos más cercanos de la imagen de test se puede utilizar diferentes medidas de distancia. En este estudio, se ha implementado cuatro medidas de distancia para realizar una comparación y posterior evaluación entre ellas. Para la descripción de las distancias se utilizará dos vectores genéricos $\vec{r} = \{r_i\}$, $i = 1, \ldots, l$ y $\vec{s} = \{s_i\}, i = 1, \ldots, l$.

1. **Distancia métrica ponderada**:

$$dist_p(\vec{r}, \vec{s}) = \left(\sum_{i=1}^{l} \omega_i \cdot |r_i - s_i|^p \right)^{\frac{1}{p}} \tag{3.8}$$

   Considerando $\omega_i = 1$, $i = 1, \ldots, l$, se obtiene la distancia de Minkowski. Se considerarán dos casos particulares: $dist_1$ (*Distancia de Manhattan*), definida con la distancia Minkowski y $p = 1$, y $dist_2$ (distancia Euclídea), con $p = 2$.

2. **Coeficiente de correlación de Pearson**. Es un coeficiente de similitud que se puede obtener como:

$$sim_{Pea}(\vec{r}, \vec{s}) = \frac{\vec{r}_d^T \cdot \vec{s}_d}{|\vec{r}_d||\vec{s}_d|} \tag{3.9}$$

   donde $\vec{r}_d = [r_1 - \bar{r}, \ldots, r_l - \bar{r}]$ y $\vec{s}_d = [s_1 - \bar{s}, \ldots, s_l - \bar{s}]$, $\bar{r} = \frac{1}{l}\sum_j r_j$, $\bar{s} = \frac{1}{l}\sum_j s_j$. Este coeficiente toma un valor en el rango $[-1, +1]$. De este valor de similitud, la distancia entre vectores puede ser definida como:

$$dist_3(\vec{r}, \vec{s}) = 1 - sim_{Pea}(\vec{r}, \vec{s}) \tag{3.10}$$

3. **Producto interno**. Se trata de otro coeficiente de similitud que puede ser calculado como el producto escalar entre los vectores a comparar:

$$sim_{cos}(\vec{r},\vec{s}) = \frac{\vec{r}^{T} \cdot \vec{s}}{|\vec{r}||\vec{s}|} \tag{3.11}$$

Como muestra la ecuación, $\vec{r}$ y $\vec{s}$ están normalizados. En ese caso, esta medida es conocida como la *similitud coseno* y toma valores en el rango $[-1,+1]$. Con ello se obtiene la distancia entre vectores como:

$$dist_4(\vec{r},\vec{s}) = 1 - sim_{in}(\vec{r},\vec{s}) \tag{3.12}$$

Por lo tanto, las cuatro medidas de distancia que se compararán en este capítulo son: $dist_1$ (distancia Manhattan), $dist_2$ (distancia Euclídea), $dist_3$ (distancia basada en la correlación de Pearson) y $dist_4$ (distancia basada en la similitud coseno).

Primero, se estudia el ratio de acierto de cada algoritmo. Este estudio muestra la habilidad del algoritmo de localización para detectar correctamente el vecino más cercano, es decir, únicamente con la información visual ser capaz de identificar correctamente la posición del modelo que geométricamente está en la posición más cercana. Las figuras 3.10–3.16 muestran el ratio de acierto. Todos los resultados están expresados en la misma escala de color.

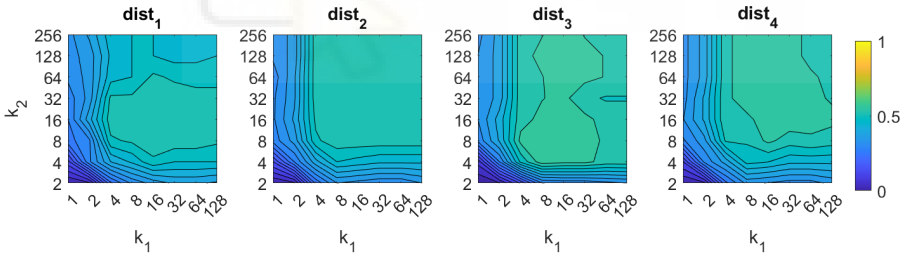


Figura 3.10: Problema de localización usando **FS**. Ratio de acierto del método. k_1 y k_2 son, respectivamente, el número de filas y de columnas en el descriptor (Cuadro 3.1).

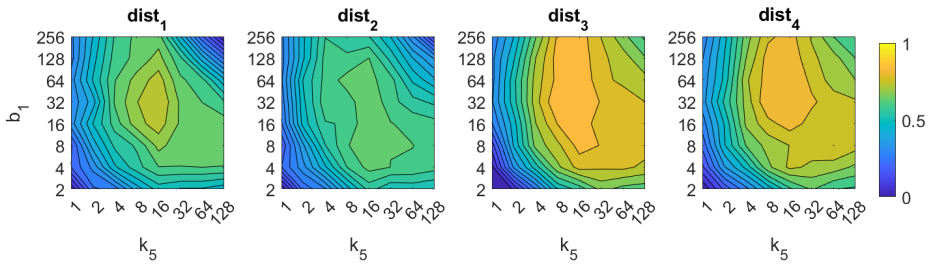


Figura 3.11: Problema de localización usando **HOG**. Ratio de acierto del método. k_5 es el número de celdas horizontales y b_1 el número de bins por histograma (Cuadro 3.1).

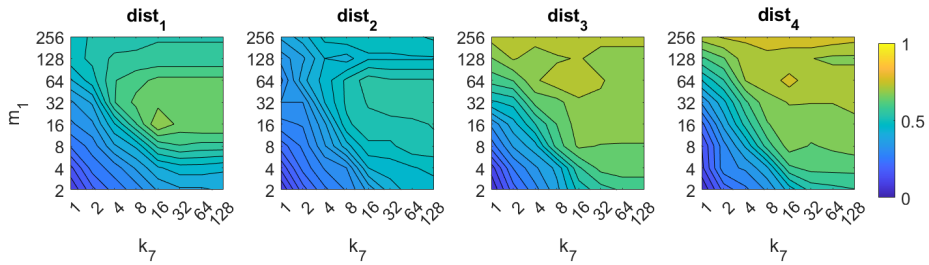


Figura 3.12: Problema de localización usando ***Gist***. Ratio de acierto del método. k_7 es el número de bloques horizontales y m_1 el número de filtros de Gabor en el descriptor (Cuadro 3.1).

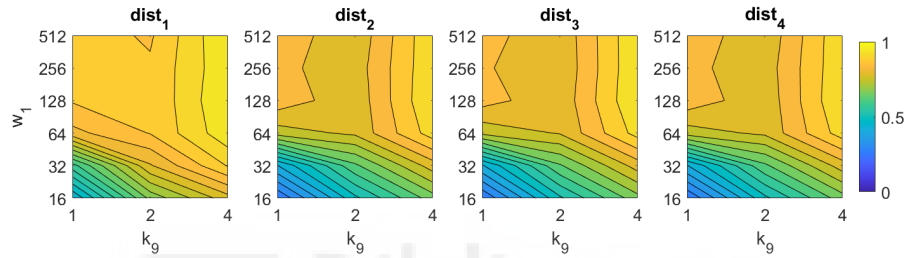


Figura 3.13: Problema de localización usando **WS**. Ratio de acierto del método. k_9 es el número de celdas horizontales y w_1 el número de ventanas por celda (Cuadro 3.1).

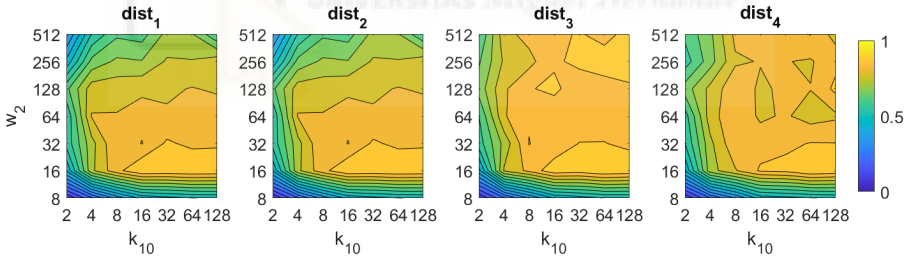


Figura 3.14: Problema de localización usando **BG**. Ratio de acierto del método. k_{10} es el número de celdas horizontaes y w_2 el número de ventanas por celda (Cuadro 3.1).

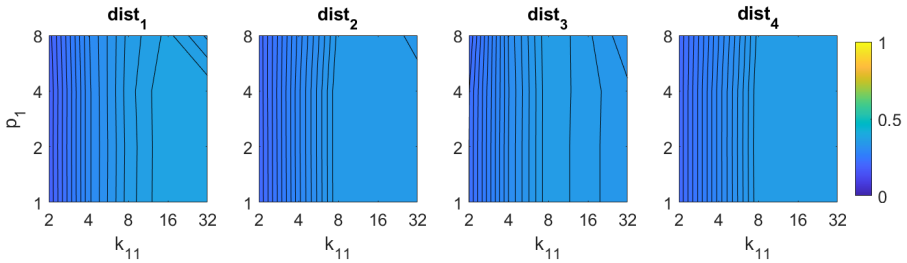


Figura 3.15: Problema de localización usando **RT–F**. Ratio de acierto del método. k_{11} es el número de filas seleccionados y p_1 es el ángulo relativo entre conjuntos consecutivos de líneas (Cuadro 3.1).

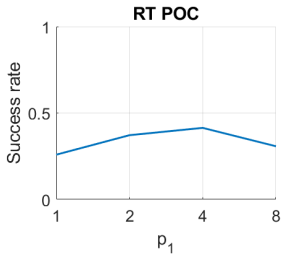


Figura 3.16: Problema de localización usando **RT–POC**. Ratio de acierto del método. p_1 es el ángulo relativo entre lineas consecutivas (Cuadro 3.1).

El comportamiento con FS cambia ligeramente dependiendo de las medidas de distancia utilizada. Los mejores resultados se obtienen con las distancias $dist_3$ y $dist_4$ con un número intermedio de filas k_1 y un número intermedio o alto de columnas k_2. En ambos casos, un número excesivamente bajo de filas y/o columnas tiene como resultado un bajo ratio de acierto del vecino más cercano. La mejor tasa de acierto es de $60\,\%$, y se obtiene con la distancia $dist_3$ y los parámetros $k_1 = k_2 = 8$.

Sobre HOG, el mejor resultado se obtiene con las distancias $dist_3$ y $dist_4$. En ambos casos, el número de celdas horizontales k_5 debe tomar un valor intermedio, rondando $k_5 = 16$. Con un alto número de celdas no se obtiene una mejor precisión. Con respecto al número de bins por histograma b_1 este debe tomar valores intermedios o altos, en los casos con $b_1 < 16$ no se obtienen resultados adecuados. En el caso de las distancias $dist_1$ y $dist_2$, un alto número de celdas y bins también proporciona malos resultados. El mejor resultado obtenido es un $89\,\%$ de acierto, este resultado se obtiene empleando la distancia $dist_3$ y $k_5 = 8, b_1 = 32$.

En el caso de *gist*, los mejores resultados se obtienen de nuevo utilizando las medidas de distancia $dist_3$ y $dist_4$. En este caso, la precisión aumenta cuando aumenta el número de máscaras m_1. Asimismo, no es necesario un alto número de celdas k_7 para obtener un buen resultado. La mejor precisión obtenida es de un $89\,\%$, esta se obtiene con la distancia $dist_3$ y los parámetros $k_7 = 32, m_1 = 256$.

En el caso del uso de *Wi-SURF*, los mejores resultados se obtienen utilizando $dist_1$ y $dist_3$. En este caso, el problema de localización se soluciona con un mejor ratio de acierto cuando se utiliza un valor alto de filas k_9 (alrededor de 4). Además, el proceso trabaja de forma adecuada con valores intermedios y altos de ventanas w_1, valores de 128 en adelante. El mejor ratio obtenido es del $97\,\%$, y se obtiene con la distancia $dist_1$ y con $k_9 = 4, w_1 = 512$.

Si se analizan los datos de *BRIEF-gist*, se observa que los mejores resultados vuelven a obtenerse con $dist_3$ and $dist_4$. Para obtener resultados adecuados es necesario un valor intermedio de celdas horizontales k_{10}, en torno a 64 celdas. Un alto número de ventanas w_2 no mejora los resultados, pero son remarcables los malos resultados obtenidos con valores bajos de k_{10} y/o w_2. El mejor resultado obtenido con BG es $93\,\%$, y se obtiene con distancia $dist_3$ y $k_{10} = 64, w_2 = 16$.

Finalmente, en el caso de *RT*, el resultado no es competitivo si se compara con el resto de descriptores de apariencia global. Por un lado, utilizando la transformada Radon y la Firma de Fourier, los mejores resultados se obtienen con las distancias $dist_1$ y $dist_4$. En este caso los parámetros no tienen gran influencia en el resultado, pero en general, con grandes valores de k_{11} y bajos valores de p_1 se obtiene los mejores ratios. Utilizando la técnica *RT–F*, el mejor resultado es de un $39\,\%$, y se obtiene con la distancia $dist_1$ y los parámetros $k_{11} = 32, p_1 = 1$. Por otro lado, utilizando el método POC, los resultados tampoco son comparables al resto de descriptores. En este caso, no hay un estudio con diferentes medidas de distancias ya que el método POC devuelve por si solo el vecino más cercano. Con este método, el mejor resultado es un $41\,\%$ de acierto y se obtiene con $p_1 = 4$.

Analizando los resultados globalmente, Wi-SURF es el descriptor que presenta un mejor ratio de acierto del vecino más cercano. El mejor ratio se obtiene con el algoritmo de Wi-SURF y la distancia $dist_1$, pero en general la distancia $dist_3$ funciona mejor que el resto de distancias en prácticamente todos los casos. HOG, *gist* y BRIEF-*gist* también son métodos con resultados aceptables. Teniendo en cuenta que el entorno suponía un gran reto debido a las condiciones cambiantes en la escena, estos descriptores de apariencia global proporcionan unos resultados remarcables.

Además del ratio de acierto del vecino más cercano, también se ha evaluado el coste computacional del proceso, para evaluar si es posible realizar la tarea de localización en tiempo real. El proceso de localización tiene que ser preciso, pero también competente en tiempos. Siguiendo esta idea las figuras 3.17–3.23 muestran el tiempo empleado para obtener el vecino más cercano, dependiendo de los parámetros del descriptor. El tamaño del descriptor depende de los parámetros del mismo. Se muestra el resultado medio del proceso con cada combinación de parámetros, con los resultados expresados en segundos. Se presentan los resultados con una escala logarítmica y así obtener una escala de color con una representación más eficiente.

Los experimentos se han llevado a cabo en un ordenador con una CPU Intel Core i7-9700® a 3 GHz y haciendo uso de la herramienta matemática Matlab®. Los tiempos resultantes no son absolutos ya que dependen del ordenador con el que trabaje el robot móvil. Pero estos resultados son comparables entre ellos porque todos han sido calculados con la misma máquina.

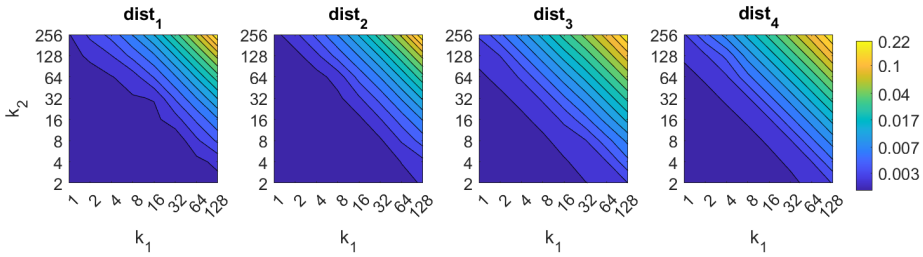


Figura 3.17: Problema de localización usando **FS**. Tiempo de computación en segundos [s].

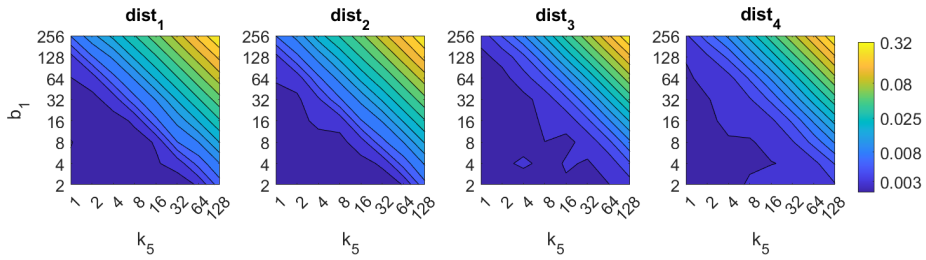


Figura 3.18: Problema de localización usando **HOG**. Tiempo de computación en segundos [s].

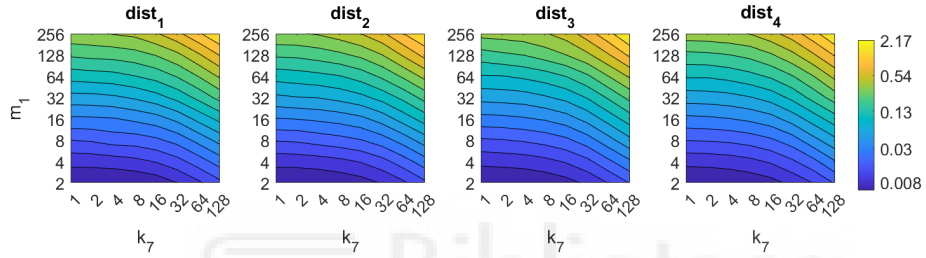


Figura 3.19: Problema de localización usando ***Gist***. Tiempo de computación en segundos [s].

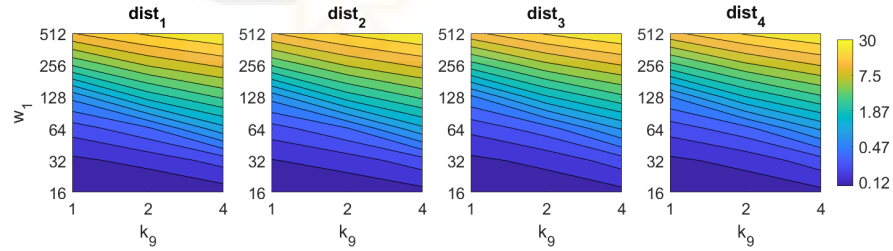


Figura 3.20: Problema de localización usando **WS**. Tiempo de computación en segundos [s].

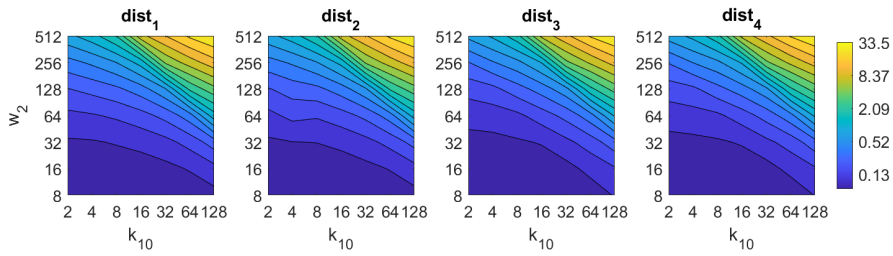


Figura 3.21: Problema de localización usando **BG**. Tiempo de computación en segundos [s].

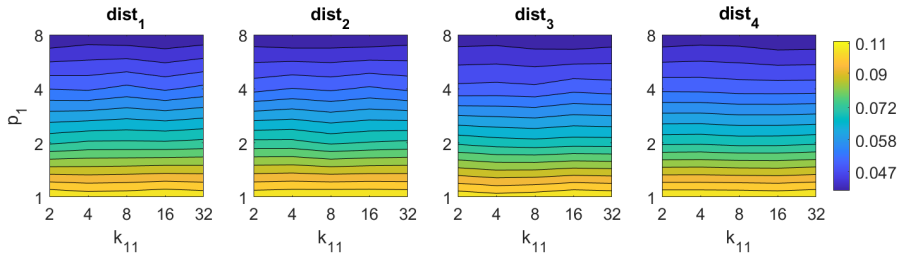


Figura 3.22: Problema de localización usando **RT–F**. Tiempo de computación en segundos [s].

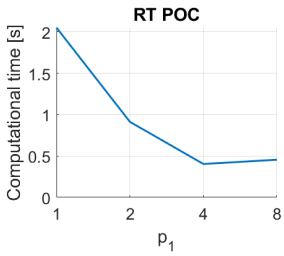


Figura 3.23: Problema de localización usando **RT–POC**. Tiempo de computación en segundos [s].

Por un lado, FS es el algoritmo más rápido. El tiempo medio para calcular el vecino más cercano cuando se obtiene una nueva *imagen de test* está por debajo de los $0{,}02$ s en la mayoría de las configuraciones. Solo cuando ambos parámetros (k_1 y k_2) toman sus valores más altos, el tiempo de computación toma valores alrededor de $0{,}22$ s. Ambos parámetros tienen influencia similar en el coste computacional. En segundo lugar, el coste computacional de HOG es ligeramente superior al de FS, y su velocidad depende de la configuración de los parámetros. Ambos parámetros b_1 y k_3 tienen una influencia similar en el tiempo de cómputo. Cuando sus valores son altos es posible obtener tiempos de $0{,}32$ s, pero con la mayoría de combinaciones el tiempo de computación es inferior a $0{,}10$ s. Evaluando el algoritmo *gist* se observa que es un método más complejo computacionalmente. El parámetro m_1 tiene una fuerte influencia en el proceso. Un alto número de máscaras junto a valores altos de celdas horizontales k_7 tiene como resultado tiempos para localizar la nueva imagen de $2{,}1$ s. Aunque es posible encontrar configuraciones que proporcionen tiempos de computación aceptables bajando el número de componentes en los parámetros.

Por otro lado, el segundo grupo de descriptores (aquellos en los cuales primero se debe girar el descriptor hasta obtener la orientación relativa entre imágenes y después se calcula la distancia entre ellos) son considerablemente más lentos. Ambos necesitan más de 2 s para sus mejores configuraciones. En Wi-SURF, w_1 tiene una fuerte influencia en el coste computacional, por ello, si es posible, es mejor no utilizar valores altos de este parámetro. Altos valores de los parámetros pueden llevar a tiempos de computación superiores a 30 s. A su vez, BRIEF-gist es todavía más pesado computacionalmente,

w_2 influye mucho en el tiempo del proceso y puede producir tiempos de cómputo de hasta $33{,}5$ s.

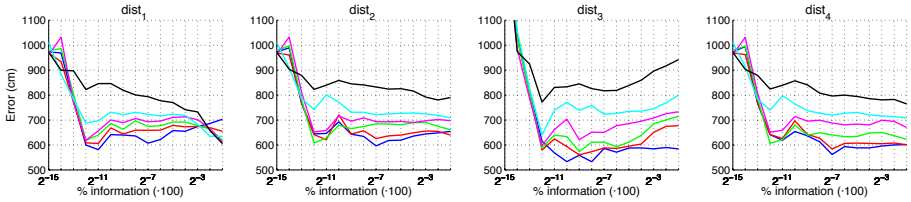
Finalmente, el método basado en la transformada de Radon y Fourier trabaja rápidamente y se obtiene resultados por debajo de los $0{,}1$ s. El método basado en Radon y POC calcula el vecino más cercano en $0{,}5$ s con valores altos p_1. A pesar de estos buenos resultados en computación, ya que los resultados basados en la transformada de Radon ofrece un ratio de acierto del vecino más cercano muy pobre, estos descriptores no se incluirán en los siguiente análisis.

3.5.2. Estimación de la posición

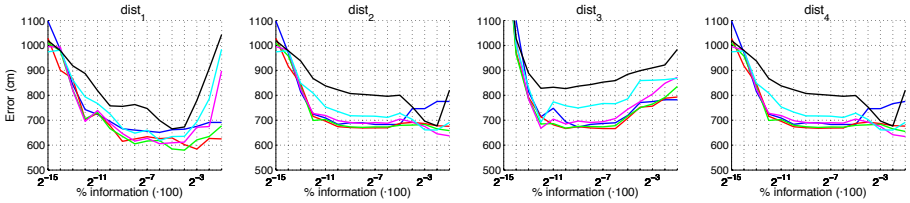
El segundo conjunto de experimentos evalúa la habilidad de los descriptores para estimar correctamente la posición del robot. Los experimentos se realizan teniendo en cuenta las alteraciones de ruido y oclusión en las imágenes de test. Los resultados de estos experimentos se presentan con el error en distancia con cada combinación frente al tamaño del descriptor. Para poder comparar descriptores. El tamaño del descriptor se expresa en porcentaje de información respecto de la cantidad de datos de la imagen original a la que describen.

En el instante t se captura una nueva imagen, para esta se calcula su descriptor y este es comparado con todos los descriptores del mapa. La comparación se realiza utilizando las distintas distancias entre descriptores. El primer vecino más cercano se asume como posición del robot. Después de detectar el vecino visual más cercano, la distancia Euclídea entre la posición del robot en el instante t y la posición del vecino más cercano se asume como error de posición. Esta distancia se puede calcular ya que se dispone de información 'Ground Truth' del lugar donde se han capturado las imágenes, pero para el proceso de localización únicamente se ha hecho uso de información visual.

Las figuras 3.24 y 3.25 presentan el error obtenido con la Firma de Fourier considerando la presencia de ruido y oclusiones. Estas figuras se subdividen en dos filas; la primera donde no se ha considerado ningún filtro (a) y una segunda donde se hace un filtrado de las imágenes con un filtro homomórfico (b). Se estudia utilizando las 1232 imágenes test y el resultado se presenta como error de posición medio y se expresa en cm. En el eje horizontal se muestra el porcentaje de información utilizado, haciendo uso de una escala logarítmica. Las etiquetas de cada representación son $\{2^{-15}, 2^{-14}, 2^{-13}, \ldots, 2^{-2}, 2^{-1}\}$ que corresponden, respectivamente, con los siguientes porcentajes de información $\{0'003\,\%, 0'06\,\%, 0'12\,\%, \ldots, 25\,\%, 50\,\%\}$. Estos porcentajes hacen referencia a la cantidad de información que contienen respecto de la imagen panorámica que describen $\left(\frac{k_1 \cdot k_2}{N_1 \cdot N_2} \cdot 100\right)$. En general, utilizando FS el uso del filtro homomórfico empeora los resultados. Como era esperado, con un alto nivel de ruido los errores son altos. Sin embargo, $dist_1$ y $dist_2$ presentan un comportamiento robusto cuando hay presencia de ruido en la imagen. Respecto de la alteración de oclusiones, el descriptor FS es bastante sensible a este fenómeno y el resultado empeora substancialmente cuando el porcentaje de oclusión aumenta.



(a)



(b)

Figura 3.24: Error medio de localización usando **FS** cuando hay ruido: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Ruido 1, — Ruido 2, — Ruido 3, — Ruido 4, — Ruido 5.

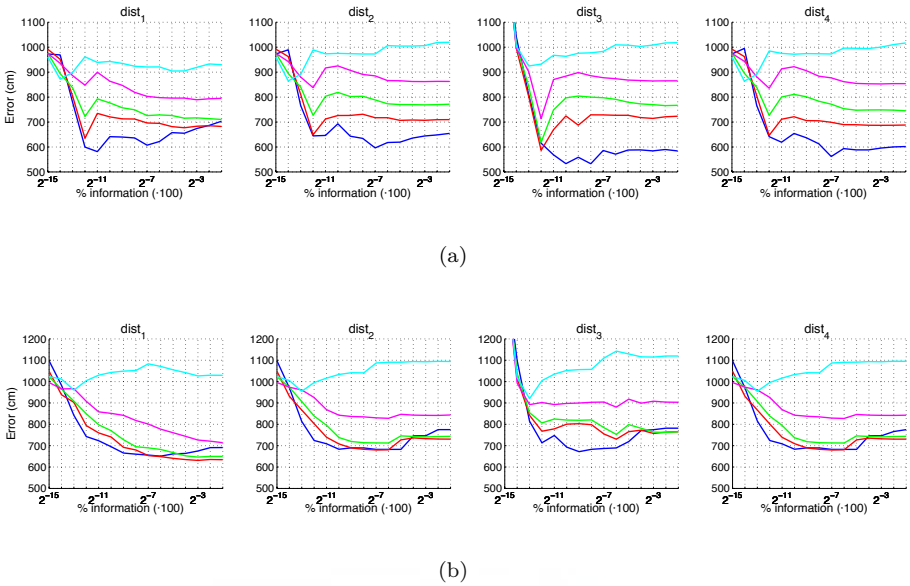
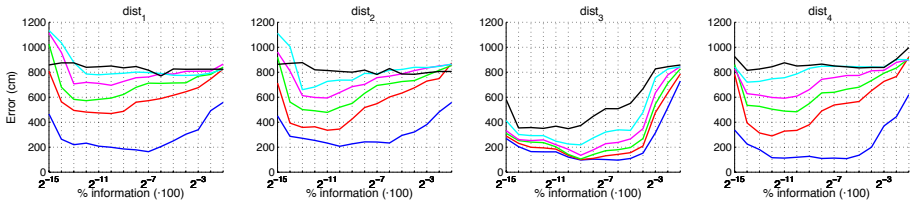
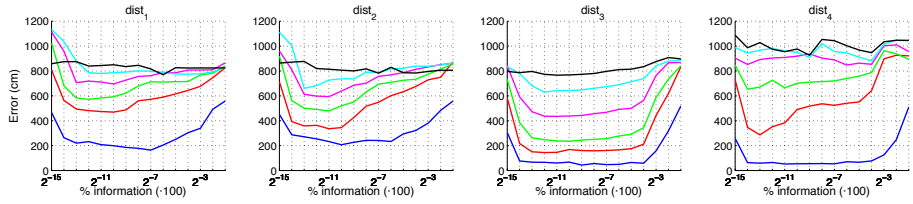


Figura 3.25: Error medio de localización usando **FS** cuando hay oclusiones: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Oclusión 1, — Oclusión 2, — Oclusión 3, — Oclusión 4.

Las figuras 3.26 y 3.27 presentan los resultados obtenidos con el Histograma de la Orientación de los Gradientes considerando, respectivamente, la presencia de ruido y oclusiones en las imágenes test. Los resultados están divididos en soluciones sin usar ningún filtro (a) y haciendo uso de filtro homomórfico (b). Como en el caso de FS, los porcentajes en el eje horizontal expresan el porcentaje de información que contiene el descriptor respecto de la imagen panorámica. En el caso de HOG se obtiene como $\left(\frac{k_5 \cdot b_1}{N_1 \cdot N_2} \cdot 100\right)$. El uso del filtro homomórfico solo mejora cuando se utilizan las distancias $dist_3$ y $dist_4$ y si hay un bajo nivel de ruido u oclusiones. Porcentajes intermedios de información tienden a presentar los mejores resultados absolutos, por tanto, no parece necesario almacenar una gran cantidad de información durante la construcción del descriptor. En presencia de ruido, el mejor resultado absoluto se obtiene con la distancia $dist_3$, sin hacer uso del filtro homomórfico y con una cantidad intermedia de información. Comparado con otros métodos de descripción, HOG destaca por su robustez ante la presencia de oclusiones en las imágenes de test.

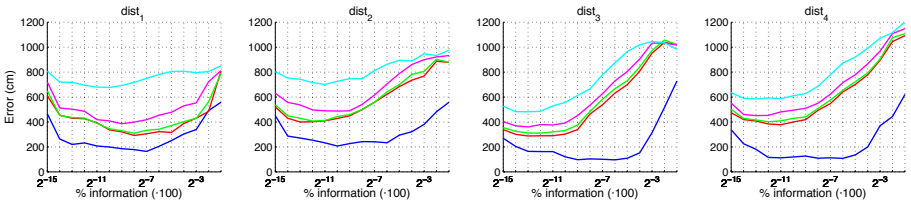


(a)

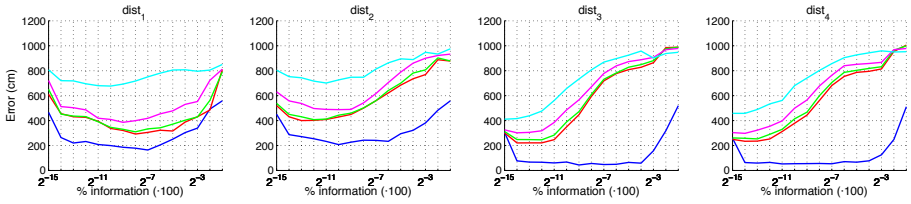


(b)

Figura 3.26: Error medio de localización usando **HOG** cuando hay ruido: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Ruido 1, — Ruido 2, — Ruido 3, — Ruido 4, — Ruido 5.



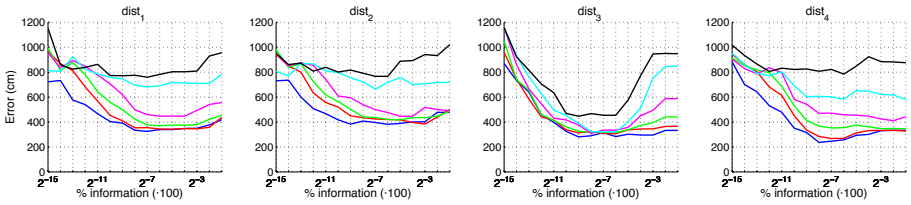
(a)



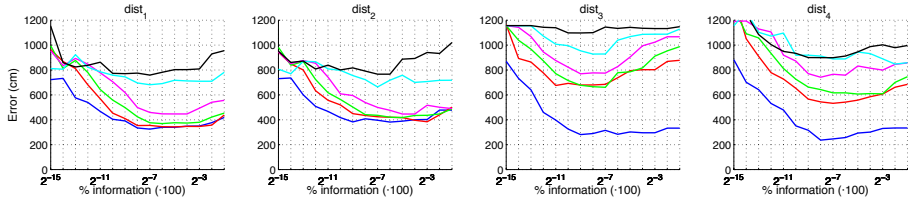
(b)

Figura 3.27: Error medio de localización usando **HOG** cuando hay oclusiones: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Oclusión 1, — Oclusión 2, — Oclusión 3, — Oclusión 4.

Adicionalmente, las figuras 3.28 y 3.29 presentan el resultado obtenido con *gist* considerando en las imágenes de test la presencia de ruido y oclusiones respectivamente. Como en los casos anteriores, la primera fila corresponde con los resultados obtenidos sin el uso de filtros (a) y la segunda fila haciendo uso del filtro homomórfico (b). El porcentaje de información que el descriptor toma de la imagen original se puede obtener como $\left(\frac{2 \cdot k_7 \cdot m_1}{N_1 \cdot N_2} \cdot 100\right)$. El uso del filtro homomórfico no mejora los resultados de localización cuando se hace uso del descriptor *gist*. De hecho, hay situaciones en las que el resultado empeora. En presencia de ruido, la distancia $dist_3$ presenta una considerable reducción del error respecto otros casos. Se obtienen buenos resultados con descriptores de tamaño intermedio, por ello, no son necesarios descriptores de tamaños muy grandes para obtener resultados aceptables.

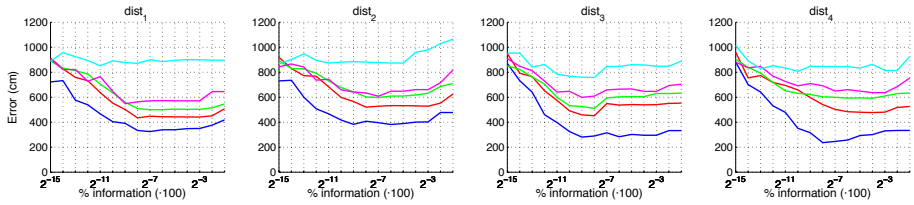


(a)

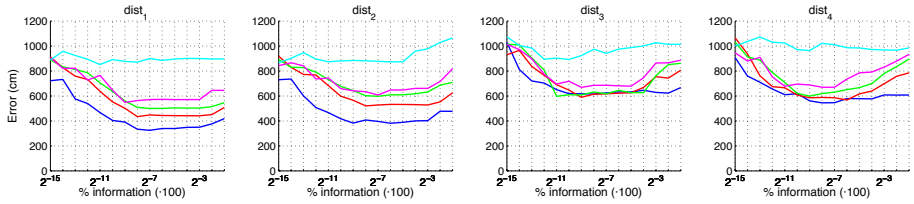


(b)

Figura 3.28: Error medio de localización usando ***Gist*** cuando hay ruido: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Ruido 1, — Ruido 2, — Ruido 3, — Ruido 4, — Ruido 5.



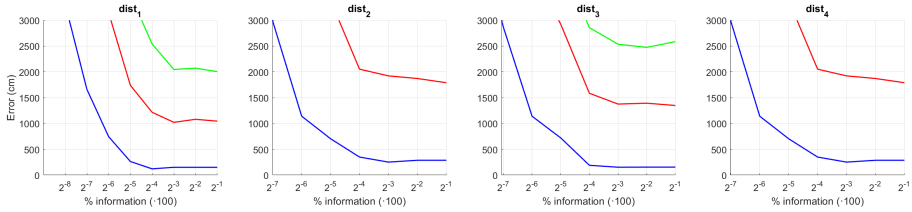
(a)



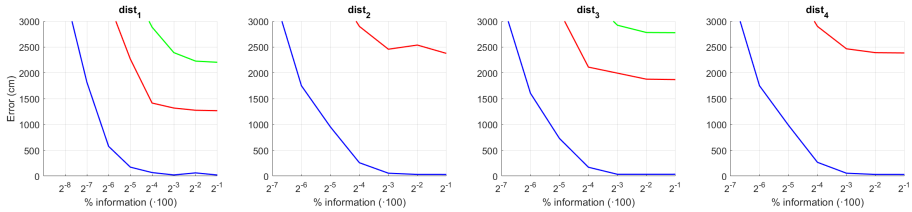
(b)

Figura 3.29: Error medio de localización usando ***Gist*** cuando hay oclusiones: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Oclusión 1, — Oclusión 2, — Oclusión 3, — Oclusión 4.

En cuarto lugar se presentan los resultados obtenidos con el descriptor *Wi-SURF*. Estos se muestran en las figuras 3.30 y 3.31, donde se presenta el error de localización medio con las distintas configuraciones cuando hay presencia de las alteraciones de ruido y oclusión. En estas figuras, primero se muestran los resultados sin hacer uso de filtros (a) y después haciendo uso de filtro homomórfico (b). En el eje horizontal se muestra la información con respecto a la imagen panorámica original $\left(\frac{k_9 \cdot w_1 \cdot 64}{N_1 \cdot N_2} \cdot 100\right)$. El uso del filtro homomórfico no reduce el error de localización. En este caso, el descriptor se ve muy influenciado por la presencia de ruido en la imagen, una presencia de estas alteraciones provoca altos valores de error. Es muy significativo que los resultados en los que la imagen test no tiene ni ruido ni oclusiones son mejores que los obtenidos con los descriptores previos, pero una vez aparece estas alteraciones el error se dispara. En general, con porcentajes de información intermedios o altos se obtienen los mejores resultados de localización. En cuanto a las distancias, los resultados más convenientes se presentan con las distancias $dist_1$ o $dist_3$.

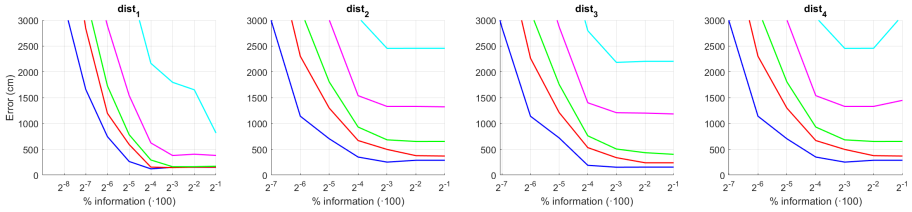


(a)

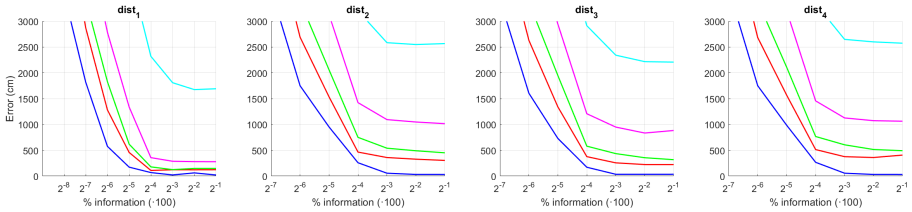


(b)

Figura 3.30: Error medio de localización usando **WS** cuando hay ruido: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Ruido 1, — Ruido 2, — Ruido 3, — Ruido 4, — Ruido 5.



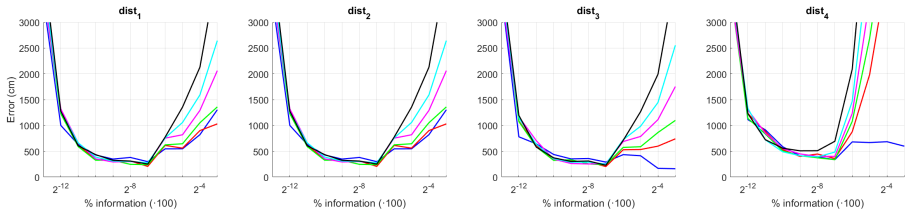
(a)



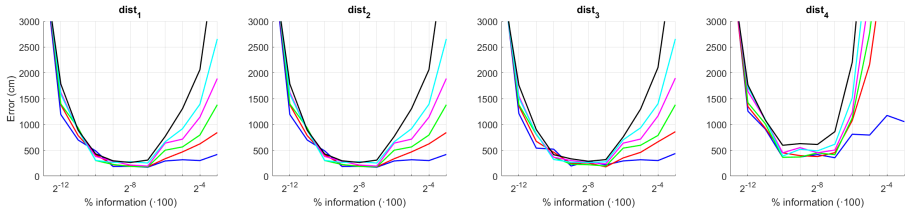
(b)

Figura 3.31: Error medio de localización usando **WS** cuando hay oclusiones: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Oclusión 1, — Oclusión 2, — Oclusión 3, — Oclusión 4.

Finalmente las figuras 3.32 y 3.33 presentan los resultados obtenidos con el método *BRIEF-gist* considerando, respectivamente, la presencia de ruido y oclusiones en las imágenes test. En una primera fila se presentan los resultados sin presencia de filtros (a) y en una segunda con el uso de un filtro homomórfico (b). Como en figuras previas, los porcentajes del eje horizontal presentan la cantidad de información de la imagen que contiene el descriptor. En el caso de BG, se puede obtener como $\left(\frac{k_{10} \cdot w_2}{N_1 \cdot N_2} \cdot 100\right)$. En este caso los mejores resultados se obtienen con cantidades intermedias de información. De este modo, no es necesario la construcción de grandes descriptores para obtener buenos resultados. En general, el filtro tiende a mejorar los resultados, esta característica es más clara cuando hay presencia de ruido. De este descriptor es muy destacable el desempeño cuando aparece la alteración de ruido, ya que esta no afecta prácticamente a los resultados. Comparado con otros descriptores, BRIEF-gist presenta un alto error en condiciones ideales, pero controla el error cuando alteraciones aparecen en la escena y obtiene buenos resultados incluso con altos porcentajes de ruido. En el caso de las oclusiones, el descriptor trabaja correctamente cuando no hay oclusiones en la imagen o estas aparece en un bajo porcentaje, pero su precisión empeora con grandes cantidades de oclusiones.

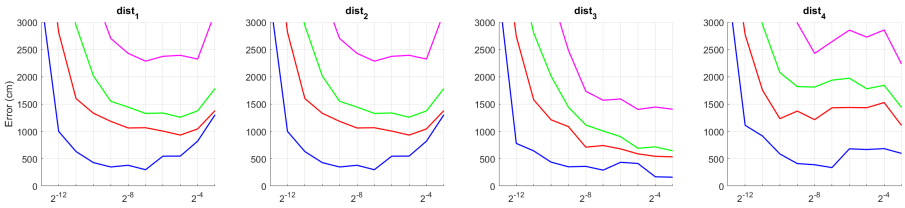


(a)

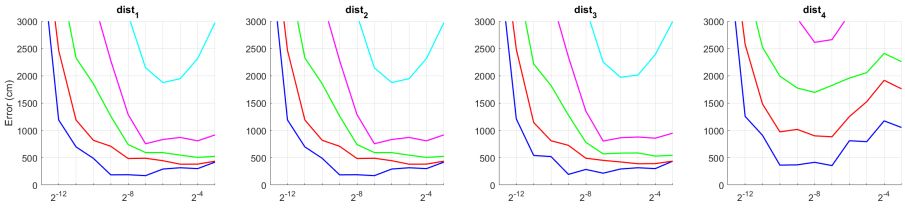


(b)

Figura 3.32: Error medio de localización usando **BG** cuando hay ruido: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: Original, Ruido 1, Ruido 2, Ruido 3, Ruido 4, Ruido 5.



(a)



(b)

Figura 3.33: Error medio de localización usando **BG** cuando hay oclusiones: (**a**) sin filtro y (**b**) con filtro homomórfico. Leyenda: — Original, — Oclusión 1, — Oclusión 2, — Oclusión 3, — Oclusión 4.

Si se analizan las figuras y los errores de localización de manera conjunta, se puede llegar a diversas conclusiones de manera general. Primero, HOG presenta buenos resultados de localización bajo condiciones ideales. Estos resultados se degradan con la presencia del ruido o las oclusiones, pero algunas configuraciones resisten a la presencia de estos fenómenos. En segundo lugar, el uso de *gist* conduce a peores resultados bajo condiciones ideales, pero es robusto ante efectos que empeoran la visibilidad de la escena, principalmente ante ruido. Además, WS proporciona los mejores resultados en condiciones ideales, sin embargo, el resultado empeora notablemente con la presencia de ruido u oclusiones. Finalmente, el resultado de BG en condiciones ideales no es remarcable, sin embargo, este descriptor presenta una alta robustez ante la presencia de ruido y oclusiones. En general, el filtro homomórfico no mejora los resultados, salvo con el uso de WS y algunas configuraciones de *gist*. Los resultados con las distintas distancias no varían mucho, aunque los mejores resultados se obtienen con las distancias $dist_1$ y $dist_3$.

3.5.3. Estimación de la orientación

En esta sección se resuelve el problema de la estimación de la orientación relativa. El problema del cálculo de la orientación se evalúa en este estudio independientemente de la estimación de la posición. Siguiendo esto, el descriptor de orientación de la imagen test se compara con el descriptor de orientación de la imagen capturada en la

posición más cercana geométricamente. El problema se resuelve utilizando los algoritmos presentados en la sección 3.3. Se exponen los resultados de todos los descriptores excepto los basado en la transformada de Radon, debido a su escaso acierto en la tarea de localización del vecino más cercano.

En primer lugar, se presentan los resultados obtenidos haciendo uso de la Firma de Fourier. La figura 3.34 muestra los resultados de orientación cuando hay presencia de ruido en las imágenes de test. El resultado se presenta como error medio de orientación obtenido tras repetir el experimento con las 1232 imágenes test. La figura muestra que el algoritmo es bastante robusto ante la presencia de ruido. La configuración óptima de los parámetros es un uso de valores intermedios o altos de filas (k_3) e intermedio de columnas (k_4). Un alto número de columnas empeora el resultado. Adicionalmente se presenta la estimación de la orientación con la presencia de oclusiones en las imágenes test en la figura 3.35. En esta figura se muestra que la influencia de las oclusiones es más alta y empeora los resultados relacionados con el aumento de oclusiones en la imagen. De todas formas, algunas configuraciones permiten obtener errores menores a 10 grados incluso con un $40\,\%$ de la imagen ocluida. Como en el caso del fenómeno del ruido, los mejores resultados se obtienen con valores intermedio o altos de filas (k_3) e intermedio de columnas (k_4). El coste computacional del cálculo de orientación se muestra en la figura 3.36, expresado en segundos. El descriptor FS es capaz de estimar la orientación relativa de manera rápida para la mayoría de las configuraciones de k_3 y k_4 y solo altos valores de ambos requiere un coste computacional mayor a $0{,}1$ s. ambos parámetros tienen una influencia parecida en el coste computacional del proceso.

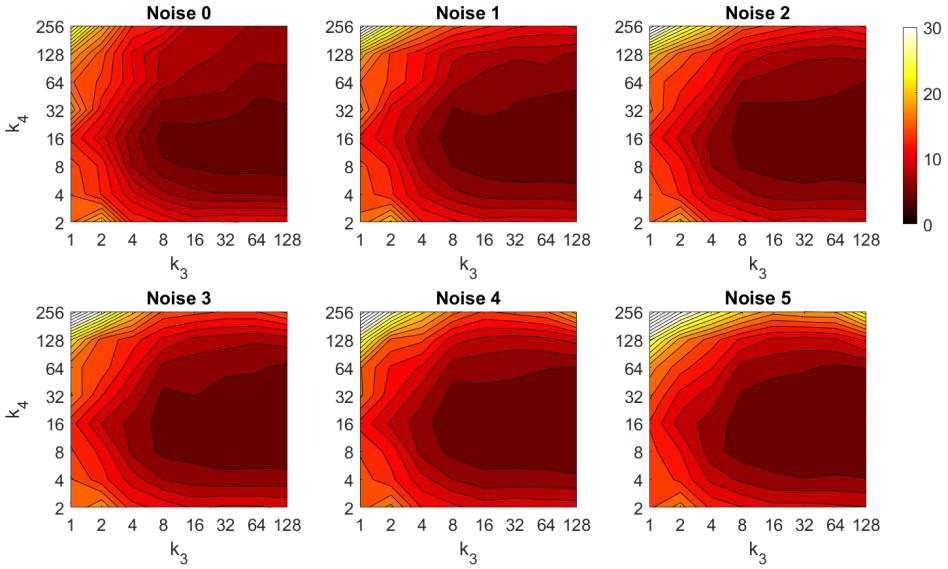


Figura 3.34: Error medio de orientación (grados) en presencia de ruido usando **FS**.

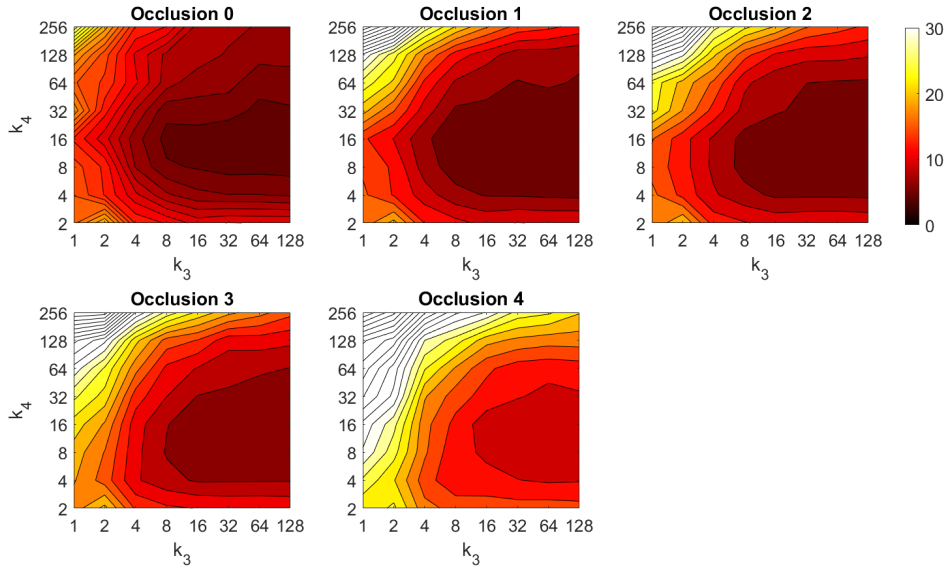


Figura 3.35: Error medio de orientación (grados) en presencia de oclusiones usando **FS**.

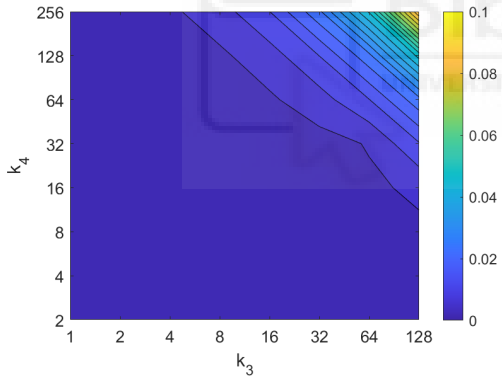


Figura 3.36: Coste computacional (segundos) en la estimación de la orientación usando **FS**.

A continuación, se presenta el análisis de los resultados obtenidos con HOG. Para visualizar los datos de error estos se presentan respecto a los parámetros l_1 (ancho de las celdas verticales en el descriptor de orientación) y d_1 (distancia entre celdas verticales consecutivas, existiendo normalmente celdas superpuestas para un mejor reconocimiento de la orientación relativa entre descriptores). La figura 3.37 muestra el error de orientación medio tras evaluar todas las imágenes de test, analizando la influencia de diferentes niveles de ruido que pueden aparecer en la imagen. En términos generales, valores intermedios y bajos de d_1 y valores altos de l_1 producen los mejores resultados (errores de orientación bajos). Además, HOG proporciona unos descriptores robustos ante la presencia de ruido, ya que, aunque aumenta, el error no lo hace

substancialmente con una mayor presencia de ruido. En general, HOG tiende a ser una mejor elección para estimar la orientación si se compara con FS. Para el análisis de la influencia de las oclusiones se puede estudiar la figura 3.38. Como en el caso de FS, la influencia de las oclusiones en la estimación de la orientación es mayor que la del ruido, y el error aumenta rápidamente cuando aparecen altos porcentajes de oclusiones en la imagen. A pesar de esto, valores altos de l_1 tienden a producir errores de orientación relativamente bajos, independientemente del nivel de oclusión. En el caso del parámetro d_1 son preferibles valores bajos de este parámetro. Finalmente, la figura 3.39 muestra el tiempo necesario para estimar la orientación en segundos. La mayoría de las combinaciones ofrecen bajos tiempos. El tiempo aumenta conforme aumenta l_1 y/o disminuye d_1, teniendo l_1 una mayor influencia en el coste computacional. Solo valores muy altos de l_1 combinados con valores bajos de d_1 tienen como resultado un valor alto del coste computacional.

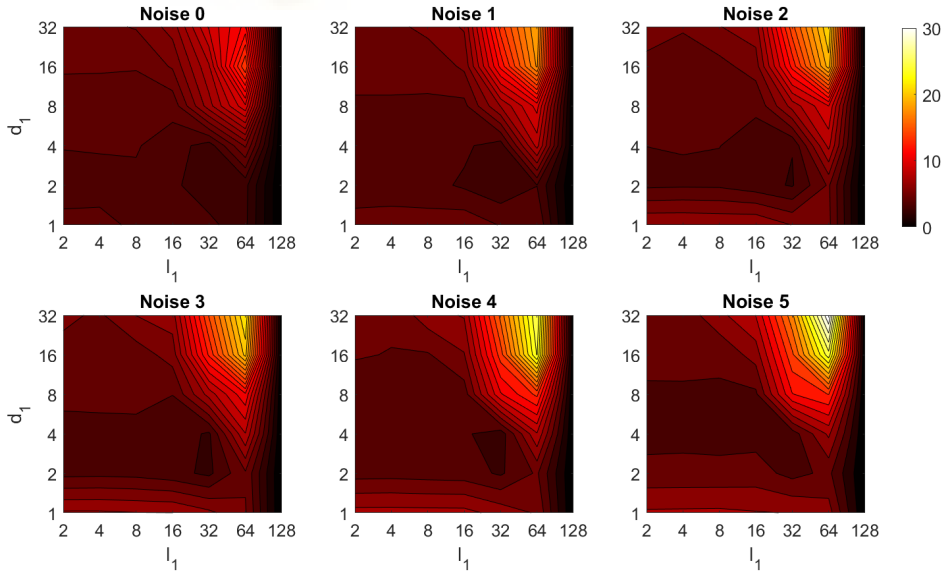


Figura 3.37: Error medio de orientación (grados) en presencia de ruido usando **HOG**.

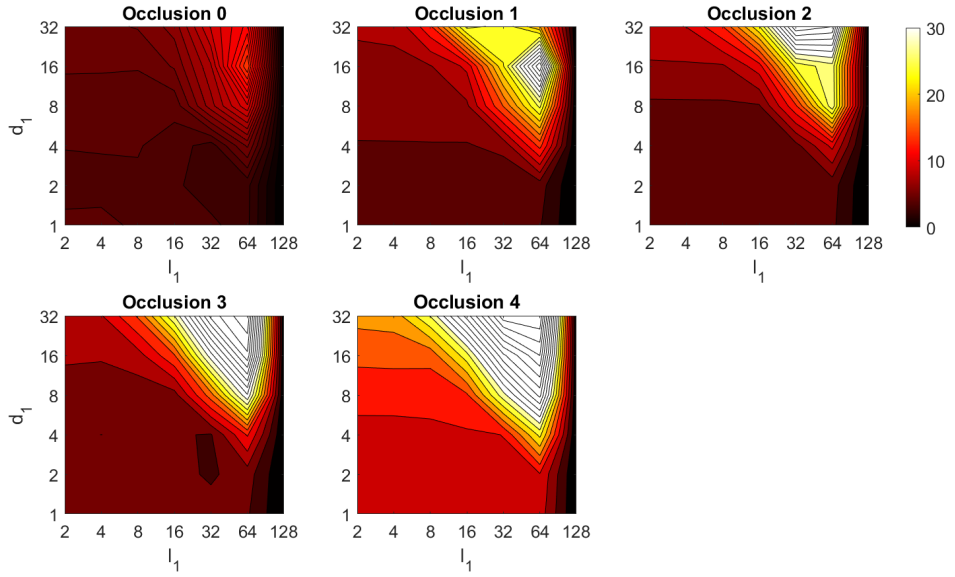


Figura 3.38: Error medio de orientación (grados) en presencia de oclusiones usando **HOG**.

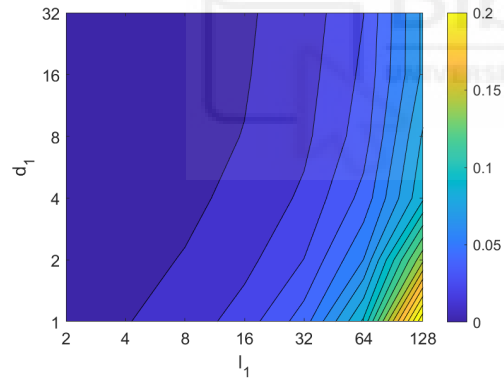


Figura 3.39: Coste computacional (segundos) en la estimación de la orientación usando **HOG**.

En tercer lugar, se presentan y comentan los resultados de error tras estimar la orientación con el descriptor *gist*. Los resultados se muestran en la figura 3.40 donde se puede estudiar el error medio de orientación (grados) al considerar diferentes configuraciones de l_2 (ancho de las celdas verticales en el descriptor de orientación) y d_2 (distancia entre celdas verticales consecutivas, existiendo normalmente celdas superpuestas para un mejor reconocimiento de la orientación relativa entre descriptores). Con el aumento del valor de d_2, el error tiende a aumentar, aunque, como en el caso de HOG, valores altos del ancho de los bloques verticales (l_2) suponen también un buen resultado independientemente del valor de d_2. El algoritmo *gist* proporciona resultados relativamente aceptables pese a la presencia de alteraciones cuando están en un bajo

porcentaje y solo un alto porcentaje de los fenómenos de oclusión y ruido empeoran el resultado. Para finalizar, también se estudia el tiempo necesario (tiempo medio en segundos) para realizar la tarea de estimación de orientación relativa. Estos resultados se muestran en la figura 3.41. Estas gráficas de tiempos muestran que d_2 es un parámetro con una influencia predominante en el coste computacional. Valores altos de este parámetro producen tiempos aceptables y bajos valores de d_2 junto a altos valores de l_2 producen tiempos de estimación de hasta $0{,}2$s.

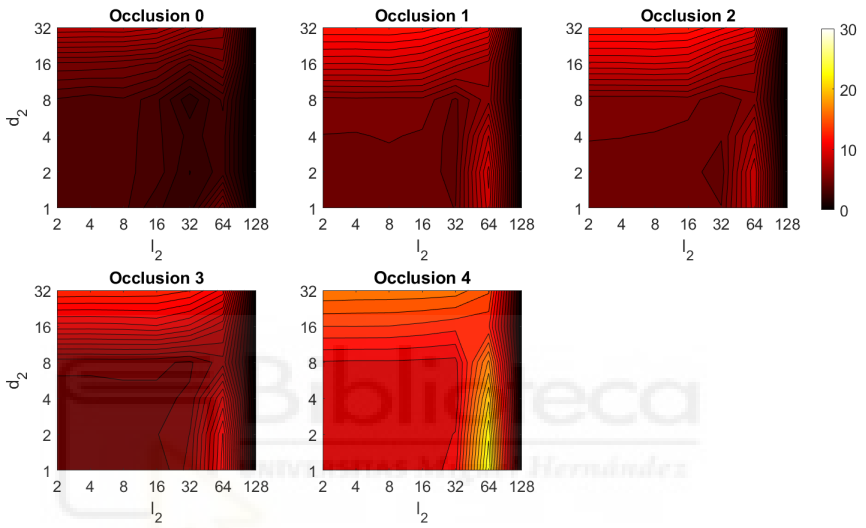


Figura 3.40: Error medio de orientación (grados) en presencia de ruido usando ***gist***.

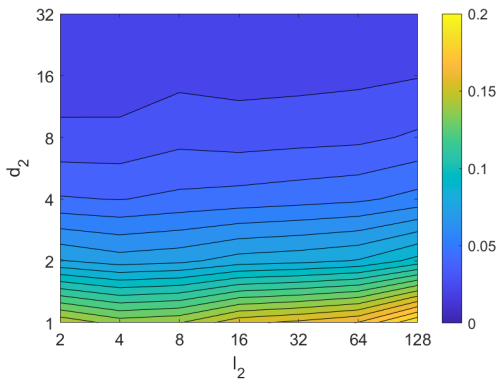


Figura 3.41: Coste computacional (segundos) en la estimación de la orientación usando ***gist***.

Además, en la figura 3.42 se puede encontrar el resultado de la estimación de la orientación relativa usando el descriptor *Wi-SURF* teniendo en cuenta el fenómeno de

ruido y la variación de los parámetros k_9 y w_1. Se muestra una fuerte influencia de las alteraciones producidas por el ruido en el resultado. Se destaca que los resultados sin ruido son aceptables (entre 5 y 10 grados), pero el error se incrementa notablemente con la aparición de ruido en la escena. Si la imagen está corrupta por un valor de varianza mayor a $\sigma^2 = 0{,}0025$, el error es siempre superior a 30 grados. La influencia producida por los niveles de oclusiones se puede chequear en la figura 3.43. En el caso de las oclusiones los datos de error son más aceptables, sin ofrecer unos resultados robustos ya que son considerablemente mayores a los resultados obtenidos con HOG y *gist*. En general, el error se tiende a optimizar con valores medios de w_1. Para finalizar con este descriptor se muestran los resultados del coste computacional en la figura 3.44. Aquí se muestra que el número de ventanas (w_1) es el que más influencia tiene en el cómputo de la estimación de orientación. Valores altos de ambos parámetros producen altos valores de coste computacional, pero estos son bajos en relación a los tiempos ofrecidos por otros descriptores.

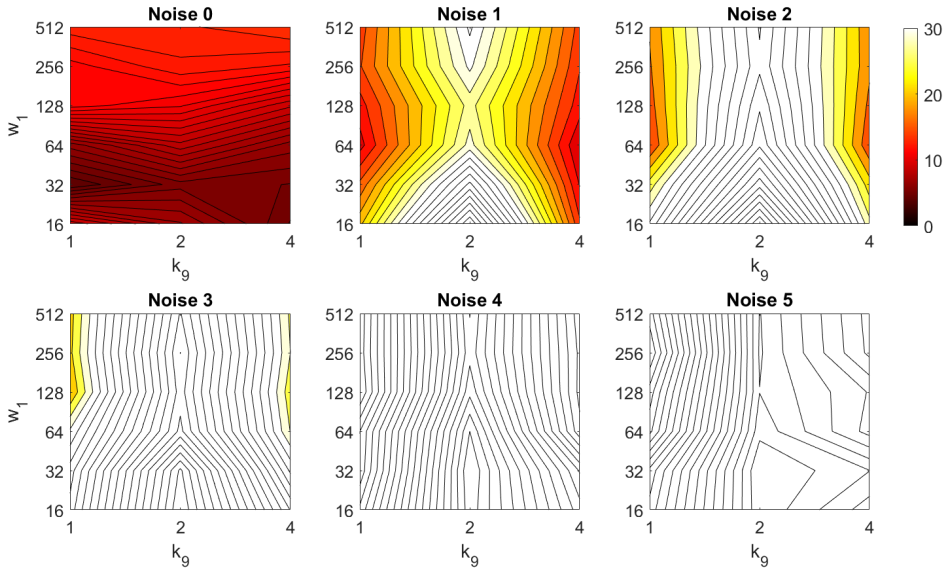


Figura 3.42: Error medio de orientación (grados) en presencia de ruido usando **WS**.

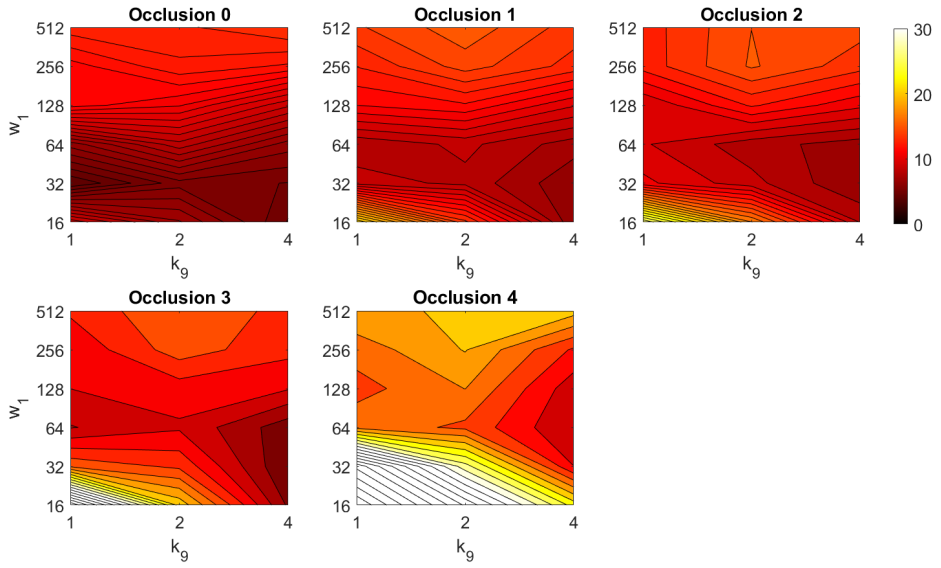


Figura 3.43: Error medio de orientación (grados) en presencia de oclusiones usando **WS**.

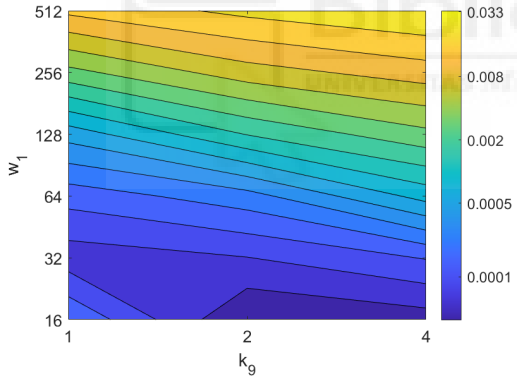


Figura 3.44: Coste computacional (segundos) en la estimación de la orientación usando **WS**.

Por último, se presentan los resultados obtenidos con el descriptor *BRIEF-gist*, considerando la variación de los parámetros w_2 y k_{10}. La figura 3.45 muestra el error medio de orientación con los diferentes niveles de ruido. En términos generales, los mejores resultados se obtienen con una combinación de valores intermedios o altos del número de celdas (k_{10}) y un número intermedio de ventanas (w_2). Un alto número de ventanas ofrece resultados negativos. Además, el algoritmo BG proporciona un descriptor bastante robusto ante la presencia de ruido en las imágenes, ya que el resultado no cambia substancialmente pese a la presencia de este fenómeno. Sin embargo, la influencia de oclusiones parciales en la imagen tiene una peor influencia a la hora de estimar la orientación del robot, esto se puede observar en la figura 3.46. Como con WS, el

algoritmo actúa considerablemente mal para altos porcentajes de oclusiones, aunque con un $5\,\%$ o $10\,\%$ de oclusión en la imagen los errores no aumentan notablemente. Como antes, valores intermedios de w_2 ofrecen los mejores resultados. Finalmente, la figura 3.47 muestra el tiempo necesario para estimar la orientación. Todas las comparaciones de w_2 y k_{10} ofrecen tiempos de computación competentes, siendo estos menores a $0{,}012$s para cualquier combinación de parámetros.

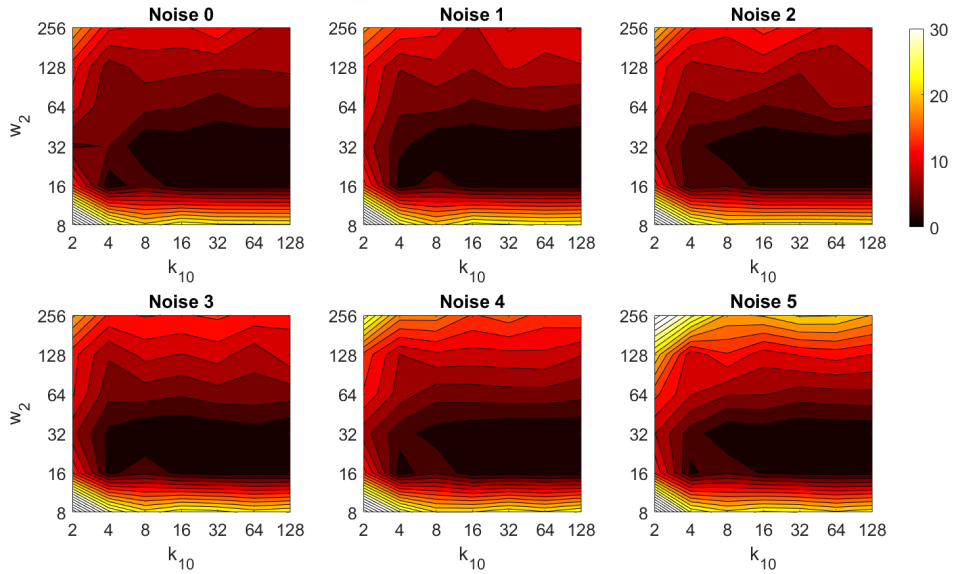


Figura 3.45: Error medio de orientación (grados) en presencia de ruido usando **BG**.

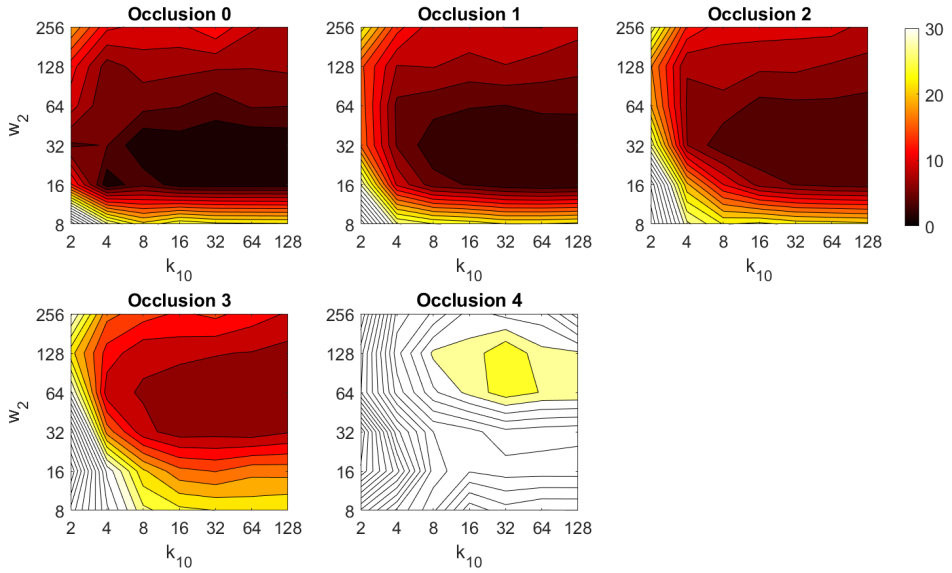


Figura 3.46: Error medio de orientación (grados) en presencia de oclusiones usando **BG**.

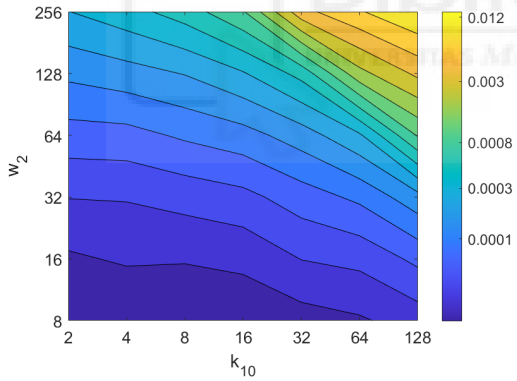


Figura 3.47: Coste computacional (segundos) en la estimación de la orientación usando **BG**.

En términos generales, HOG y *gist* producen relativamente mejores resultados en la estimación de la orientación relativa, aunque aparezcan fenómenos adversos como ruido u oclusiones en la imagen y solo grandes porcentajes de estas alteraciones empeorarán notoriamente el proceso. Además, existen diferentes configuraciones en las que se ofrece un buen balance entre error y coste computacional. Por otro lado, el descriptor BRIEF-gist ofrece errores aceptables, aunque aparezcan grandes niveles de ruido en la imagen y solo grandes porcentajes de oclusiones empeoran notablemente el desempeño adecuado del algoritmo y todo ello con un coste computacional muy bajo. Por último, el uso de Wi-SURF solo se recomienda con condiciones ideales ya que el

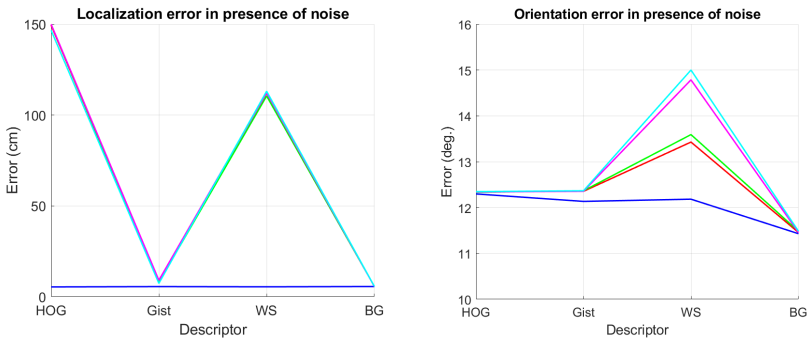
error de orientación crece remarcablemente con la presencia de oclusiones o ruido en la imagen.

3.5.4. Evaluación con imágenes tomadas en una trayectoria

Para concluir con la sección de experimentos, se lleva a cabo una localización con imágenes extraidas de la base de datos COLD [135]. Esta base de datos es pública y contiene conjuntos de imágenes que han sido capturadas mientras el robot móvil realiza una trayectoria en un entorno de interior. Por tanto, el resultado de esta sección permite evaluar el desempeño de los descriptores en un entorno diferente y con un conjunto de datos capturado a lo largo de una trayectoria real.

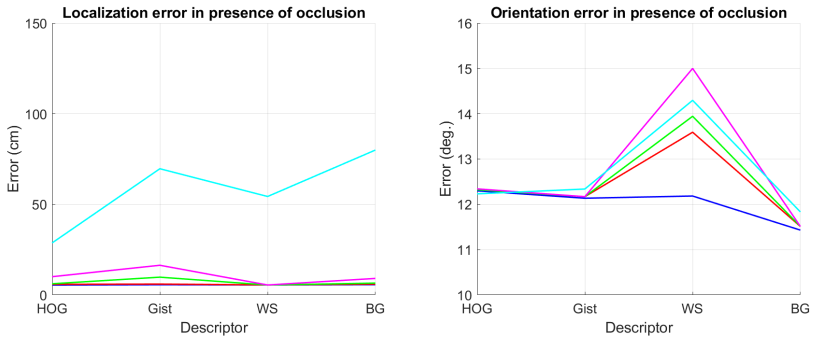
Para llevar a cabo este experimento, se ha escogido la base de datos de Saarbrücken [135]. Para crear el conjunto de entrenamiento se ha seleccionado una trayectoria con imágenes capturadas, de media, cada 30 cm. Por otro lado, en la trayectoria de test las imágenes están separadas 6 cm. Las imágenes de test se utilizan para solucionar el problema de localización. Para solucionar este problema se sigue el proceso descrito en la sección 3.3.

Los resultados se presentan en las figuras 3.48 y 3.49. Como en el resto de experimentos presentados en este capítulo, se estima tanto la posición como la orientación relativa del robot considerando efectos como el ruido y oclusiones. Los descriptores que se incluyen en este experimento son HOG, *gist*, WS y BG, ya que han mostrado un buen desempeño en los experimentos anteriores. Adicionalmente, sus parámetros más relevantes están sintonizados con los valores que proporcionaron las mejores estimaciones en los subapartados anteriores. Los niveles de ruido y oclusiones son las mismas que las incluidas en los experimentos previos: presencia de ruido Gaussiano ($\sigma^2 = \{0, 0{,}0025, 0{,}05, 0{,}01, 0{,}02\}$) y oclusiones parciales considerando el ($\{0, 5, 10, 20, 40\}\,\%$) de la imagen.



(a) (b)

Figura 3.48: Error medio con la trayectoria de Saarbrücken en presencia de ruido. (**a**) Error medio de posición (cm)y (**b**) error medio de orientación (grados). Leyenda: — Original, — Ruido 1, — Ruido 2, — Ruido 3, — Ruido 4.



(a) (b)

Figura 3.49: Error medio con la trayectoria de Saarbrücken en presencia de oclusiones. (**a**) Error medio de posición (cm)y (**b**) error medio de orientación (grados). Leyenda: — Original, — Oclusión 1, — Oclusión 2, — Oclusión 3, — Oclusión 4.

Primero, la figura 3.48 muestra (a) el error medio en la tarea de localización (expresada en cm) y (b) el error medio en la tarea de estimación de la orientación relativa (expresada en grados). En este experimento se consideran diferentes niveles de ruido. Después, la figura 3.49 muestra el mismo experimento pero considerando diferentes niveles de oclusiones parciales de la imagen.

Es importante destacar que estos errores no son directamente comparables con el error absoluto presentado en las subsecciones previas, ya que el conjunto de imágenes es diferente y las imágenes fueron tomas desde puntos de captura diferentes. No obstante, estas gráficas permiten evaluar el desempeño de los descriptores con un conjunto de datos capturado a lo largo de una trayectoria y saber si los descriptores presentan tendencias similares en diferentes tipos de entornos y conjuntos de datos.

La figura 3.48a muestra que el error cometido por los diferentes descriptores al calcular la posición relativa en condiciones ideales (es decir, sin ruido) es bastante similar. Además, se puede extraer que *gist* y BG resisten bastante bien a la presencia de ruido. Sin embargo, HOG y WS empeoran notoriamente su precisión cuando el nivel de ruido crece. Estos resultados están en la linea de los presentados en esta sección. Respecto de la orientación relativa con ruido, la figura 3.48b muestra que HOG, *gist* y BG son bastante robustos, mientras que WS da peores resultados con altos niveles de ruido.

La figura 3.49a muestra que los cuatro métodos de descripción presentan relativamente buenos resultados en presencia de oclusiones, excepto para altos niveles de oclusión. En este caso, HOG es el descriptor con mejor rendimiento. Respecto a la tarea de estimación de la orientación relativa en presencia de oclusiones, la figura 3.49b muestra que HOG, *gist* y BG tienen un buen rendimiento, independientemente del nivel de oclusión, pero WS aumenta rápidamente el error con altos niveles de oclusión. Estos resultados están en sintonía con los obtenidos para una localización en modo de rejilla.

3.6. Conclusiones

Este capítulo se ha centrado en el estudio del problema de localización utilizando un modelo visual previamente construido que represente el entorno. El problema se ha resuelto como una tarea de localización visual absoluta. Haciendo uso de un sensor de visión catadióptrico montado en el robot móvil se estima la posición y orientación del robot. Para extraer información relevante de las imágenes se hace uso de los descriptores de apariencia global, los cuales proporcionan un único vector que describe toda la imagen. Se realiza una evaluación comparativa de seis familias de descriptores de apariencia global. Además, se tienen en cuenta distintos efectos negativos que pueden ocurrir en tareas de localización reales como puede ser la aparición de ruido en las imágenes o de oclusiones que no permitan ver por completo la escena, realizando el estudio en condiciones ideales y cuando estos efectos aparecen en la captura de información.

La principal contribución de este trabajo es el estudio exhaustivo de diferentes técnicas de descripción global (FS, HOG, *gist*, WS, BG y RT) y la adaptación de algunas de estas familias de algoritmos para poder usarlos con escenas panorámicas de las cuales poder describir información de posición y de orientación. Durante la tarea de localización absoluta, el robot detecta su posición más cercana y posteriormente estima su orientación relativa respecto a la imagen más cercana. Adicionalmente, el coste computacional del proceso también se ha estudiado, incluyendo la influencia de cada uno de los parámetros que tienen posibilidad de ser modificados durante el proceso de descripción.

El estudio revela que los algoritmos de FS y RT presentan un coste computacional bajo, igual que ciertas configuraciones específicas de HOG y *gist*, pero los descriptores Wi-SURF y BRIEF-gist son menos competitivos computacionalmente hablando. Desde este punto de vista, FS, RT, HOG y *gist* son más útiles para tareas en tiempo real.

Adicionalmente, se ha estudiado la tarea de localización con los diferentes descriptores. Primero, el estudio se ha centrado en el problema de localización, tratando de encontrar el vecino visual más cercano en el modelo. Todos los descriptores han sido testeados haciendo uso de diferentes medidas de distancia y los resultados muestran que los descriptores Wi-SURF y BRIEF-gist presentan un mejor ratio de acierto que el resto de métodos. Los métodos HOG y *gist* también ofrecen unos resultados relativos bastante adecuados con ciertas combinaciones de parámetros, teniendo HOG la mejor relación entre coste computacional y ratio de acierto en la localización. Por último, los descriptores FS y RT no ofrecen resultados competentes en la tarea de localización.

Por otro lado, se realiza una segunda tarea de localización en la cual se estima la posición y orientación del robot y se calcula el error de estimación. Dentro de este estudio se corroboran cuatro aspectos: (a) los descriptores HOG y *gist* presentan un resultado competente bajo condiciones ideales, pero además son bastante robustos ante ruido y oclusiones, en especial el descriptor HOG; (b) el descriptor Wi-SURF proporciona los mejores resultados en condiciones ideales pero el efecto que tienen los fenómenos adversos negativos en la imagen influye mucho en la tarea de localización, obteniendo

resultados de error poco asumibles; (c) BRIEF-gist es el descriptor más robusto ante efectos adversos, pero su calidad no es remarcable en condiciones ideales y (d) los descriptores FS y RT no son adecuados para solventar una tarea de localización con las características de nuestro estudio. En resumen, los mejores resultados en condiciones ideales se obtienen usando el descriptor WS, pero ante fenómenos visuales adversos, como podría ser el ruido o las oclusiones es mejor hacer uso de los descriptores HOG y *gist*.

Los resultados expuestos han demostrado que los métodos de apariencia global son un enfoque viable para llevar a cabo la tarea de localización. Gracias a ellos, el robot es capaz de construir un modelo del entorno y usar este mapa base para estimar la posición y orientación del robot en el entorno con precisión, además con un coste computacional relativamente eficiente. Este hecho puede tener implicaciones interesantes en el desarrollo de nuevos algoritmos dentro del campo de la robótica móvil. Como ejemplo, este concepto se puede usar para construir mapas híbridos que almacenan la información en diferentes capas, con diferentes grados de precisión en cada capa: en una primera capa de alto nivel se permite obtener una estimación rápida, aunque imprecisa, de la posición del robot y en una capa de bajo nivel contiene información con precisión geométrica que permite al robot refinar la estimación de posición. Los métodos de apariencia global se pueden usar por si solos o como combinación con otras técnicas basadas en la detección de características y de esta manera crear algoritmos más eficientes.

Todas estas conclusiones animan a profundizar en el marco de la visión para tareas de robótica móvil autónoma. Para obtener un sistema autónomo capaz de construir un mapa del entorno y localizarse en él de manera adecuada es necesario seguir trabajando en esta línea considerando diferentes trabajos futuros. Primero, es posible crear un proceso de captación de imágenes de manera autónoma y poder obtener una representación más óptima del entorno. Segundo, también es necesario realizar esta tarea de evaluación de manera más profunda en tareas de localización reales con cambios visuales del entorno propias de ambientes dinámicos donde pueda aparecer cambios en la iluminación. Tercero, es posible estudiar los procesos de mapping y localización visual para integrarlos en un sistema topológico de SLAM que lleve a cabo tanto la creación del modelo como la localización desde cero. Para la optimización de estos algoritmos, también se considera en trabajos futuros realizar una comparación completa entre las técnicas de apariencia global, las basadas en características locales y las nuevas técnicas de estimación de pose basadas en inteligencia artificial.

3.7. Publicaciones relacionadas con este capítulo

Los principales resultados presentados en este capítulo están relacionados en la siguientes publicaciones:

- V. Román, L. Payá, A. Peidró, M. Ballesta, O. Reinoso. The Role of Global Appearance of Omnidirectional Images in Relative Distance and Orientation.

Sensors. Ed. MDPI. 21(10), 3327 (Mayo 2021). [146] **JCR-SCI Impact Factor: 3.576**, Quartile **Q1**.

  - Este artículo presenta el análisis de los descriptores de apariencia global descritos en este capítulo. Las imágenes han sido captadas con cámaras omnidireccionales en la Universidad Miguel Hernández. Se realiza una comparación exhaustiva de las técnicas para la tarea de estimación de la posición y de orientación, así como su robustez ante la aparición de defectos como ruido y oclusiones.

- V. Román, L. Payá, S. Cebollada, A. Peidró, O. Reinoso. Evaluating the Robustness of New Holistic Description Methods in Position Estimation of Mobile Robots. Lecture Notes in Electrical Engineering. Informatics in Control, Automation and Robotics (2022) - 793 (207-225). Ed. Springer [141].

  - Este artículo presenta un estudio de cómo es posible modificar los descriptores de apariencia global para obtener unos vectores que optimicen las tareas de localización. Se trata de una serie de modificaciones en las operaciones o en la forma de construir los vectores con la finalidad de una continua mejora del proceso. Se trabaja sobre imágenes omnidireccionales proporcionadas por la base de datos COLD.

- V. Román, L. Payá, S. Cebollada, A. Peidró, O. Reinoso. An Evaluation of New Global Appearance Descriptor Techniques for Visual Localization in Mobile Robots under Changing Lighting Conditions. ICINCO 2020: 17th Intl. Conf. On Informatics in Control, Automation and Robotics (Streaming Online, 7-9 julio 2020). Ed. INSTICC [144].

- V. Román, L. Payá, M. Flores, S. Cebollada, O. Reinoso. Performance of New Global Appearance Description Methods in Localization of Mobile Robots. Robot 2019, Advances in Intelligent Systems and Computing (2019) - 2 (351-363) Ed. Springer [145].

- V. Román, L. Payá, O. Reinoso. Evaluating the robustness of global appearance descriptors in a visual localization task, under changing lighting conditions. ICINCO 2018, 15th International Conference on Informatics in Control, Automation and Robotics (Oporto, Portugal, 29-31 agosto, 2018) Ed. INSTICC [147].

  - Este conjunto de publicaciones en actas de congresos recoge los diferentes trabajos de optimización de los descriptores para realizar tareas de localización. Los descriptores de apariencia global tienen varios parámetros y peculiaridades en la construcción que se han conseguido ajustar con los experimentos expuestos en estos trabajos. Los trabajos se han realizado con imágenes omnidireccionales de la base de datos COLD y con las imágenes tomadas en la Universidad Miguel Hernández. La experimentación permite mejorar los resultados cuando hay presencia de cambios de iluminación en el entorno.

4 Creación de modelos de entornos interiores de manera incremental

4.1. Introducción

La tarea de mapeo o mapping es, junto a la localización, una de las tareas primordiales para la navegación de un robot móvil autónomo. Como se ha mencionado en el Capítulo 1, los sensores de visión y en especial las cámaras omnidirecionales son una alternativa fiable para la realización de estas tareas. El éxito en resolver problemas de mapeo viene dado por el continuo progreso en las habilidades de percepción y computación por parte de las máquinas, que han facilitado su operabilidad en grandes entornos heterogéneos. Dos grandes marcos han sido utilizados para la resolución de tareas de mapping: los mapas métricos y los mapas topológicos. Los mapas métricos son aquellos en los que se expresan relaciones espaciales entre entidades. Este tipo de mapas proporciona las coordenadas de cada objeto respecto al sistema de referencia global con precisión geométrica [42, 114]. Los mapas topológicos son representaciones relacionales [180], es decir, en el mapa se muestra la relación de conectividad que existe entre las diferentes entidades, por ejemplo nodos y enlaces en los que se indica que hay una adyacencia espacial entre las diferentes localizaciones. Una alternativa eficiente para la organización de información es mediante los mapas jerárquicos, que incluyen una jerarquía de capas (métricas, topológicas o ambas), con diferentes niveles de granularidad precisión [36, 80].

Los mapas jerárquicos constituyen una alternativa conveniente para almacenar la información. En este tipo de mapas, la información se distribuye en varias capas, con diferentes niveles de granularidad, que permiten resolver la tarea de localización con distintos niveles de precisión. De esta manera, las capas de alto nivel permiten estimar la posición de forma gruesa mientras que en las capas más profundas la localización se

va afinando. Una importante iniciativa para construir mapas jerárquicos es el uso de técnicas clustering, como el *clustering espectral* [100]. Muchos son los ejemplos del uso de métodos de clustering espectral para construir mapas topológicos con información visual [55, 175, 161, 129]. Estas técnicas han sido muy usadas por su buen desempeño para agrupar datos de muy alta dimensión y su uso está justificado si se compara con otros métodos conocidos [191, 33].

Pese a los aceptables resultados que habitualmente se han obtenido con el uso del clustering espectral, el problema de este método en su uso como herramienta para construir mapas de manera incremental es triple. Primero, muchos de los métodos de clustering espectral requieren conocer el número de nodos finales previamente. Además, realizar cálculos entre sus entidades puede ser costoso en términos de coste computacional. Este problema se va acrecentando conforme la cantidad de datos se hace grande. Finalmente, y como consecuencia del problema previo, no es posible realizar un clustering espectral de manera *on-line* cuando la cantidad de datos está creciendo constantemente, es decir, cuando el robot se está moviendo constantemente y capturando nuevas imágenes que deben ser introducidas al modelo.

Debido a estos problemas, es preciso el desarrollo de métodos de clustering incremental, como los trabajos de Valgren *et al.* [174] en los que se presenta un método de construcción de mapas topológicos de manera incremental. Para ello hacen uso del descriptor local SIFT y crean una aplicación que va aumentando el número de nodos continuamente mientras el robot explora el terreno. De acuerdo con [174], crear una aplicación de mapping incremental es una buena idea si se cumple una serie de criterios: (a) Cuando la información no se puede representar en un espacio n-dimensional de manera fácil, pero sí que es posible de calcular la similitud entre individuos. (b) Cuando los datos a programar tienen un alto coste computacional y una aproximación a los resultados puede ser aceptada, el clustering incremental es más rápido y permitirá hacer tarea *on-line*. (c) Y finalmente, cuando el número de nodos final es desconocido, los métodos de clustering incremental no necesitan que se les prefije un número de nodos finales, sino que el resultado variará dependiendo de una serie de umbrales.

Los criterios mencionados se cumplen en nuestro trabajo ya que la similitud entre descriptores de apariencia global se puede calcular; el método empleado no necesita calcular una matriz de afinidad, lo que facilita el cálculo *on-line* y el número de clusters más adecuado no se conoce de antemano. Siguiendo esta idea, se presenta una herramienta para realizar mapeos de manera incremental. Esta herramienta permite a un robot móvil construir un mapa o modelo e ir actualizando la información de este mientras explora un entorno desconocido y va capturando información, en nuestro caso imágenes.

Por todo ello, en este capítulo se presenta el trabajo enmarcado en las tareas de clustering incremental con el diversas comprobaciones de cierre de ciclos. El objetivo es crear mapas topológicos y jerárquicos de manera incremental, utilizando para estas tareas únicamente información visual. Esta propuesta usa imágenes omnidireccionales y los descriptores de apariencia global, en concreto, se utilizan lo descriptores HOG y

gist. Cada conjunto de imágenes visualmente similares se agrupan en un único nodo, y a su vez cada cluster contará con vector representativo asociado. Este método no requiere conocer el número de clusters previamente, por el contrario se definen diferentes parámetros cuyo ajuste dará como resultado condiciones más o menos restrictivas para fomentar la creación de nodos. Adicionalmente, el método propuesto se puede utilizar de manera *on-line*, ya que actualiza el modelo cada vez que el robot toma una nueva imagen.

Para verificar un método de mapping visual resulta importante demostrar que es un método robusto en entornos heterogéneos y dinámicos. En operaciones con condiciones reales, un robot autónomo debe ser capaz de lidiar con eventos como cambios de iluminación en las escenas, escenarios parcialmente ocluidos por personas, objetos y/u otros robots o cambios en la posición de los muebles entre otras cosas. Por esta razón, los experimentos han sido realizados con conjuntos de imágenes capturados durante horas de trabajo y en diferentes emplazamientos, todos ellos considerados escenarios dinámicos y heterogéneos.

El resto del capítulo está estructurado de la siguiente manera. La sección 4.2 hace un repaso de los descriptores de apariencia global utilizados para realizar la tarea de mapeo. Después de esto, la sección 4.3, explica el método de mapping incremental propuesto. En la sección 4.4 se muestra la información importante sobre los entornos y las bases de datos, las herramientas y el equipo de adquisición utilizados durante los experimentos. Los resultados de estos experimentos son mostrados en la sección 4.5. A continuación, la sección 4.6 se describe nuevas versiones del algoritmo que suponen una mejora de la tarea de mapping. La sección 4.7 resume las conclusiones de estos trabajos. Además, la sección 4.8 presenta, a modo de tablas suplementarias, un resumen de los diferentes parámetros utilizados en el proceso. Finalmente, la sección 4.9 presenta las publicaciones relacionadas con el presente trabajo.

4.2. Revisión de los descriptores de apariencia global

En esta sección, se repasan los métodos de descripción de información visual de las imágenes. Para realizar esta labor, se va a hacer uso de los descriptores de apariencia global, vistos en gran profundidad en el capítulo anterior (Capítulo **??**. Descriptores de Apariencia Global en tareas de Localización). Primero, la información es captada por una cámara omnidireccional y la imagen captada se transforma a panorámica. Por tanto, el punto de inicio es una imagen $i(x,y) \in \mathbb{R}^{N_x \times N_y}$ y tras el uso de los descriptores de apariencia global se reduce a un descriptor unidimensional $\vec{d} \in \mathbb{R}^{t \times 1}$ donde t es el tamaño del descriptor.

Existe una gran variedad de descriptores de apariencia global, pero para realizar la tarea de creación de mapas se va a hacer uso de HOG y *gist*. Tras los experimentos realizados hasta la fecha en localización, los descriptores de HOG y *gist* son los que han ofrecido un rendimiento superior ante efectos adversos según el capítulo **??**. Principalmente, ambos descriptores tienen en común que para la construcción del vector descriptor es necesario dividir la imagen en una serie de celdas. Dependiendo del número

de celdas, la forma y otros parámetros de construcción el descriptor tendrá tamaños diferentes. Haciendo uso de las diferentes formas que las celdas pueden tomar, para este trabajo, el descriptor se va a construir usando dos formatos. Un método en el que las celdas son horizontales, no solapadas y están uniformemente distribuidas [126] y un segundo método que consiste en construir el vector con celdas verticales con algo de solapamiento entre celdas consecutivas [145]. Estas versiones ya han sido utilizadas en el capítulo anterior. La primera es la forma más conocida de construir el descriptor de posición mientras que la segunda sirve para construir descriptores que permitan estimar la orientación. Sin embargo, el modo de uso de la segunda versión es diferente en este caso, ya que también se usará como descriptor de similitud entre escenas. Para poder usar estos vectores como descriptores de similitud, primero se debe detectar la diferencia de orientación entre los descriptores; una vez detectado este giro, la distancia entre descriptores puede ser empleada como medida de similitud. La combinación de ambas medidas de información constituye un descriptor de similitud entre imágenes invariante a la orientación. La figura 3.2 muestra las dos ideas de construcción de estos descriptores.

Durante este trabajo, los descriptores calculados con las celdas horizontales son conocidos como descriptores de posición, mientras que los descriptores construidos con celdas verticales se los denomina descriptores de orientación. La finalidad de utilizar ambos tipos de descriptores de manera conjunta es doble. Por un lado, se tiene en cuenta la información obtenida por pura información posicional del robot y se complementa con la influencia que puede tener la orientación de este. Por otro lado, esta idea puede ayudar al sistema a reducir el *aliasing perceptual*, que ocurre cuando dos imágenes captadas en localizaciones diferentes tienen información visual semejante. Combinar información obtenida de bloques horizontales y de bloques verticales puede proporcionar resultados más fiables en lo que respecta al matching entre imágenes.

Una vez explicados los diferentes métodos de selección de celdas, se presentan los descriptores empleados. El proceso se ha realizado empleando descriptores basados en el Histograma de la Orientación de los Gradientes (HOG) y basados en *gist*. La construcción de estos descriptores está explicada de forma extensa en el capítulo anterior (Capítulo3. Descriptores de Apariencia Global en tareas de Localización), aunque en las siguientes subsecciones se realiza una breve revisión de estos métodos. Asimismo, el cuadro 4.5 presenta los parámetros generales empleados en la creación de los descriptores y el cuadro 4.4 resume los parámetros con impacto en el tamaño del descriptor.

4.2.1. Histograma de la Orientación de los Gradientes (HOG)

Esencialmente, el proceso consiste en calcular el gradiente de la imagen y obtener el módulo y argumento de este. Después de esto, utilizando las diferentes celdas descritas antes es posible construir los descriptores de apariencia global. Al final, la información se recoge en bins dependiendo de la orientación del gradiente y se pondera cada valor por la magnitud del gradiente. Cada celda tiene su propio histograma asociado y al final el descriptor se construye concatenando todos los histogramas. Para

construir el descriptor, el número de bins y celdas debe estar definido. Estas especificaciones están especificadas en la sección 4.5.1.

Clasicamente, este método se construye dividiendo la imagen en celdas horizontales. Autores como Cebollada et al. [33] o Román et al. [147] han usado esta técnica clásica de HOG para realizar tareas de localización. Como se ha destacado en esta sección, para este trabajo de mapping se va a hacer uso de celdas horizontales (con propósitos de encontrar información posicional) y celdas verticales superpuestas (con propósito de encontrar información de orientación). Ambas propuestas están descritas de manera profunda en el capítulo anterior.

Una vez se realiza la operación de HOG toda la imagen se reduce a un descriptor cuyo tamaño depende del número de celdas y de bins. Durante este capítulo se hace referencia a los parámetros con la siguiente nomenclatura: en el descriptor de posición, b_{hp} hace referencia al número de bins por histograma y k_{hp} es el número de celdas en las que se divide la imagen. De esta forma se obtiene un descriptor $\vec{d_p} \in \mathbb{R}^{b_{hp} \cdot k_{hp} \times 1}$. En el descriptor de orientación, $dist_{ho}$ hace referencia a la distancia entre celdas consecutivas, k_{ho} es el número total de celdas, y se calcula como $N_y/dist_{ho}$, donde N_y es el número de columnas en la imagen panorámica. b_{ho} es el número de bins por histograma. Al final se construye el vector HOG para orientación $\vec{d_o} \in \mathbb{R}^{b_{ho} \cdot k_{ho} \times 1}$. Los parámetros empleados están recogidos en el cuadro 4.4 y las especificaciones en la sección 4.5.1.

4.2.2. Gist

El descriptor basado en gist se construye a partir de información de intensidad obtenida tras aplicar diferentes filtros de Gabor a diferentes niveles de resolución. Para reducir el volumen de información cada imagen filtrada se divide en un conjunto de celdas, y se calcula la intensidad media de cada celda. A continuación, se concatena la información y se recoge en el descriptor *gist*. Esta práctica se describe de manera profunda en el capítulo anterior. Los parámetros del descriptor se especifican en la sección 4.5.1.

Como se ha explicado, se llevan a cabo dos modalidades de construcción, dependiendo de la forma de las celdas. Ambas propuestas están descritas en el inicio de esta sección. Una vez se tiene la imagen a diferentes resoluciones, el algoritmo filtra las imágenes con cada una de las máscaras y divide las escenas resultantes en celdas de las que se calcula la intensidad media. Durante este capítulo se hace referencia a los parámetros con la siguiente nomenclatura: en el descriptor de posición, m_{gp} indica el número de orientaciones del filtro de Gabor, k_{gp} hace referencia a el número de celdas horizontales en las que la imagen ha sido dividida mientras que r_{gp} es el número de niveles diferentes de resolución de la imagen, y se obtiene así el descriptor $\vec{d_p} \in \mathbb{R}^{r_{gp} \cdot m_{gp} \cdot k_{gp} \times 1}$. En el descriptor de orientación, $dist_{go}$ es la distancia entre celdas consecutivas, este parámetro está relacionado con el número de celdas verticales k_{go} ya que $k_{go} = N_y/dist_{go}$, siendo N_y el número de columnas de la imagen panorámica. m_{go} es el número de orientaciones de los filtros de Gabor y r_{go} indica el número de resoluciones de la imagen. Con ello se obtiene el descriptor de orientación *gist* $\vec{d_o} \in \mathbb{R}^{r_{go} \cdot m_{go} \cdot k_{go} \times 1}$. Los parámetros empleados están recogidos en el cuadro 4.4 y las especificaciones en la sección 4.5.1

4.3. Mapas incrementales jerárquicos

En esta sección se presenta el método propuesto para crear mapas incrementales. La propuesta de este trabajo es obtener un mapa jerárquico creado incrementalmente, donde zonas similares son detectadas, compactadas y representadas como pertenecientes al mismo nodo o cluster. Esta tarea es llevada a cabo al clusterizar imágenes con características similares. En términos generales, el mapa jerárquico se construye de tal manera que cuando un grupo de nuevas imágenes no pertenece a un nodo visitado anteriormente, se crea uno nuevo con ellas; al mismo tiempo, el método es capaz de detectar cuándo un nodo es revisitado o cuándo dos nodos son lo suficientemente parecidos para unirse y crear un único nodo. El mapa jerárquico es actualizado cada vez que el robot captura una nueva imagen o grupo de imágenes. Para la implementación del método se hace uso de varios parámetros. Para facilitar la lectura se incluye el cuadro 4.5 con la información de los parámetros utilizados durante la explicación y el cuadro 4.6 con los parámetros ajustables durante el proceso de mapping.

El proceso se estructura en dos etapas de cierre de bucle. Estas etapas son conocidas como 'Node Level Loop Closure' (Cierre de ciclo en nivel nodo) e 'Image Level Loop Closure' (Cierre de ciclo en nivel imagen). En la primera, el proceso selecciona los nodos candidatos con los que la nueva imagen puede cerrar el bucle. Después, en la siguiente etapa el proceso detecta la imagen más similar de entre las imágenes pertenecientes a los nodos candidatos.

El proceso comienza con la captura de una imagen I_q y el cálculo de sus descriptores. A continuación se evalúa la nueva imagen con los nodos actuales $N^C = \{N_1, N_2..., N_C\}$ en la etapa Node Level Loop Closure (subsección 4.3.1). De entre los nodos actuales se obtienen aquellos con los que es probable que la nueva imagen cierre ciclo. Estos posibles candidatos se representan por el conjunto N^*. De este modo, si un nodo N_i cumple con la condición de cierre de bucle, N_i es elegido como candidato a cerrar el bucle y pasa a formar parte del conjunto N^*. Después de recuperar el conjunto de nodos candidatos a los que la imagen I_q puede pertenecer, la etapa Image Level Loop Closure (subsección 4.3.2) es evaluada con las imágenes que pertenecen a los nodos candidatos I^{N^*}. Un esquema del proceso se puede observar en la imagen 4.1.

Para llevar a cabo estas etapas se emplean tanto los descriptores de posición como de orientación. De cada imagen I_q se calculan ambos descriptores, y estos tienen que ser capaces de seleccionar adecuadamente el nodo con posibilidad de pertenencia. En este punto los descriptores de posición y orientación son comparados únicamente con las imágenes de los nodos candidatos en la etapa Image Level Loop Closure $I^{N^*} = \{\cup_{N_i \in N^*} I^{N_i}\}$, donde I^{N_i} son el conjunto de imágenes pertenecientes al nodo N_i y N^* el conjunto de nodos seleccionados en la etapa Node Level Loop Closure. Si la etapa Image Level Loop Closure se lleva a cabo con buenos resultados, se recupera una imagen única como más similar. Si la imagen recuperada I_i cumple con la Condición de Prominencia (subsección 4.3.3) y con la Condición de Centroide (subsección 4.3.4), I_q se añade al correspondiente nodo N_i. La figura 4.1 muestra un esquema que resume gráficamente el proceso de selección de nodo al que la nueva imagen pertenece.

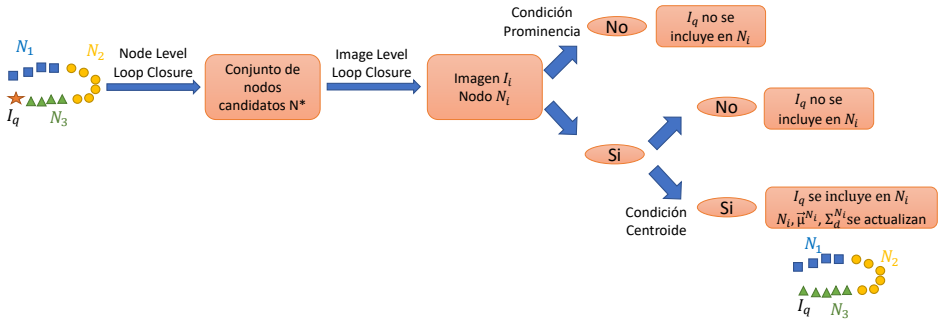


Figura 4.1: Esquema gráfico del método propuesto para procesar una imagen nueva y decidir si se incorpora a un nodo previo.

Adicionalmente, es posible que ningún nodo sea seleccionado en la etapa Node Level Loop Closure como candidato a cerrar el bucle. En ese caso, N^* se convierte en un conjunto vacío ($N^* = \emptyset$). En este caso, la nueva imagen no es incluida en ninguno de los nodos existentes. Ocurre lo mismo cuando una de las nuevas imágenes no cumple las condiciones de prominencia y/o la de centroide. Cuando un conjunto de imágenes consecutivas no son asignadas a ningún nodo se crea un nuevo cluster, expandiendo de esta forma el mapa jerárquico, cuyo número de nodos se incrementa. El resto de subsecciones describen con más detalle el proceso llevado a cabo.

4.3.1. Node Level Loop Closure

En este método se evalua la similitud de la nueva imagen I_q con cada uno de los nodos creados. Los nodos son clusters que representan zonas compactas del entorno y contienen imágenes visualmente similares. Cada imagen esta descrita con las operaciones presentadas en la sección 4.2. Utilizando los descriptores de posición, cada nodo N_l es representado por un descriptor medio $\vec{\mu}^{N_l}$ y una matriz de covarianza $\sum_d^{N_l}$.

La similitud entre la nueva imagen y un nodo se evalúa usando la distancia de Mahalanobis [104]. Siendo $\vec{d_q}$ el descriptor de la imagen I_q, la distancia $\Delta_{\vec{d_q}}^{N_l}$ entre el descriptor $\vec{d_q}$ y el nodo N_l puede ser calculada como:

$$\Delta_{\vec{d_q}}^{N_l} = (\vec{d_q} - \vec{\mu}^{N_l})^T (\sum_d^{N_l})^{-1} (\vec{d_q} - \vec{\mu}^{N_l}) \tag{4.1}$$

donde i = 1,2,...,C y C es el numero actual de nodos.

Para decidir si el nodo evaluado es candidato a cerrar el bucle, la *distancia de Mahalanobis* tiene que satisfacer la condición de similitud presentada en la ecuación (4.2), donde $\mu_{ns}^{N_l}$ y $\sigma_{ns}^{N_l}$ son la media y la desviación estandar de una distribución Gaussiana. Al construir cada nodo, el 80 % de las imágenes se usan para modelizar el nodo creando con ellas el descriptor medio $\vec{\mu}^{N_l}$ y la matriz de covarianza $\sum_d^{N_l}$. El otro 20 % de imágenes se usan para construir una distribución Gaussiana de las distancias

imagen a su propio nodo. $\mu_{ns}^{N_l}$ y $\sigma_{ns}^{N_l}$ representan la distancia media y la desviación estandar de ese 20 % de imágenes a su nodo. En la ecuación (4.2) también aparece el parámetro x. Este parámetro se trata de una variable ajustable y cuyo valor influirá en el número y constitución de los nodos. Si x tiene un valor bajo, la condición de pertenencia a alguno de los nodos creados será más difícil de satisfacer. El estudio del método ha demostrado que es más conveniente tomar valores de x altos cuando hay pocos nodos e ir reduciendo este valor conforme el número de nodos aumente. De esta manera cuando, por la cantidad de nodos existentes, la posibilidad de elección es alta, el método asegurará que solo los nodos con más coincidencia sean candidatos. Por tanto, el valor de x se reduce conforme el número de nodos aumenta, llegando a un límite en que no se hará más pequeño el valor. Este límite es una variable ajustable por el usuario a la que se ha llamado Ω y que está en el cuadro de parámetros ajustables 4.6. Siguiendo esta explicación y el parámetro Ω, x permanecerá constante con valores $\{1.7, 1.85, 2, 2.15, 2.3\}$ cuando $C \geq 9, C \geq 8, C \geq 7, C \geq 6$ o $C \geq 5$ respectivamente. El valor que toma x respecto a la cantidad de nodos descritos se puede observar en la figura 4.2.

$$\left|\Delta_{\vec{d}_q}^{N_l} - \mu_{ns}^{N_l}\right| \leq \left|x\sigma_{ns}^{N_l}\right| \tag{4.2}$$

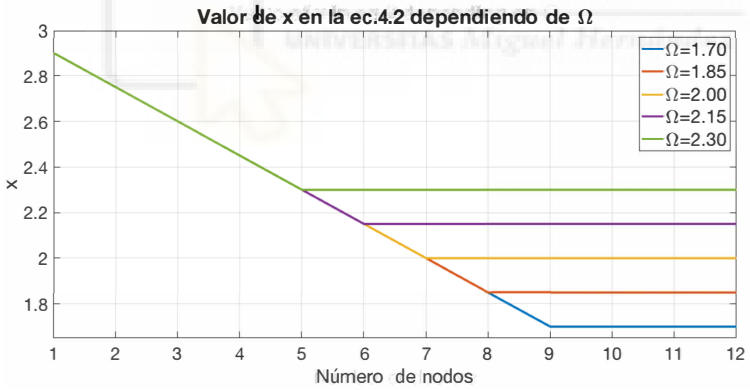


Figura 4.2: Valores de x en la ecuación (4.2) de la etapa Node Level Loop Closure Valores respecto al número de nodos.

Al finalizar la etapa Node Level Loop Closure, los nodos que satisfacen la ecuación (4.2) se consideran candidatos y se introducen en el conjunto N^*. A continuación, la etapa Image Level Loop Closure se aplica para seleccionar la imagen más parecida de entre aquellas imagenes que pertenecen a los nodos candidatos. En alguna ocasión Node Level Loop Closure no será capaz de devolver ningún candidato, en ese caso se devuelve un conjunto N^* vacío y se considera que la nueva imagen no cierra el bucle con ninguno de los nodos anteriores.

4.3.2. Image Level Loop Closure. Descriptores de posición y orientación

Esta etapa se activa una vez la etapa Node Level Loop Closure concluye satisfactoriamente con un conjunto de nodos candidatos. Este algoritmo determinará qué imagen o conjunto de imágenes cierran el bucle por ser más similares a I_q. Dado un conjunto de nodos candidatos N^*, todas sus imágenes correspondientes (I^{N^*}) son evaluadas para obtener la imagen I_i más parecida a I_q, ahorrándose el proceso la comparación con las imágenes de aquellos nodos visualmente distintos. Este problema se soluciona utilizando dos técnicas, los descriptores de posición y los de orientación, descritos en la sección 4.2.

Por un lado, se obtiene la información de posición comparando el descriptor de posición de la imagen I_q con los descriptores de posición de las imágenes (I^{N^*}). Para ello, se utiliza la *distancia Euclídea* (Ecuación (4.3)). Por otro lado, se usa la información de orientación. En esta técnica primero es necesario estimar la orientación relativa entre cada candidato I_i e I_q; a continuación, la imagen panorámica I_i y su descriptor asociado son girados la cantidad necesaria para que ambas imágenes tengan la misma orientación relativa. Una vez los descriptores tienen la misma orientación se calcula la distancia entre ellos utilizando la *distancia Euclídea* (Ecuación (4.3)), donde $\vec{d_q}$ es el descriptor obtenido de la nueva imagen I_q y $\vec{d_k}$ es el descriptor de cada imagen I_k que contiene el conjunto N^* tras la etapa Node Level Loop Closure.

$$dist_{eucl}{}^{\vec{d_k}}_{\vec{d_q}} = \sqrt{\sum_{j=1}^{t} ((\vec{d_q}(j) - \vec{d_k}(j))^2} \tag{4.3}$$

Este proceso se repite para todas las imágenes I_k contenidas en el conjunto N^*. Tras calcular estas distancias, el método determina un valor de similitud con la operación inversa de los valores de distancia, de este modo las imágenes que tenían un valor de distancia bajo (imágenes visualmente más parecidas) tendrán un valor alto de similitud (ec. (4.4)). A continuación, los datos se normalizan para que la suma del valor de similitud sea igual a 1. Por último, los valores de similitud de posición y orientación se multiplican obteniendo un valor de similitud final que combina ambos tipos de información. La imagen I_i, seleccionada como aquella que cierra el bucle, es aquella que tiene un valor de similitud final mayor en el proceso de Image Level Loop Closure (ec. (4.4)), y su nodo asociado como el nodo seleccionado para I_q. Para comprender estas medidas de similitud, es posible observar la figura 4.3, donde la subfigura 4.3(a) muestra la medida de similitud entre I_q y las imágenes de los nodos candidatos, usando los descriptores de posición; la subfigura 4.3 (b) muestra también la similitud pero usando el descriptor de orientación y la subfigura 4.3 (c) muestra la medida de similitud final obtenida tras multiplicar las medidas de posición y orientación. Como se puede observar en las subfiguras, ciertas imágenes no tienen valor de similitud, esto es debido a que sus nodos asociados no habían superado la etapa Node Level Loop Closure y no pertenecían a (I^{N^*}) . En el caso concreto de la figura 4.3, solo los nodos

C, I, J habían superado esta etapa.

$$sim^{\vec{d}_k}_{\vec{d}_q} = \frac{1}{dist_{eucl}{}^{\vec{d}_k}_{\vec{d}_q}} \tag{4.4}$$

$$i = \arg\max_k(sim^{\vec{d}_k}_{\vec{d}_q}) \tag{4.5}$$

Si ningún nodo había sido seleccionado ($N^* = \emptyset$) la etapa Image Level Loop Closure no puede ser ejecutada. En ese caso, a la nueva imagen I_q no se le asigna ningún nodo y permanece coom no-clasificado, esperando más información de las siguientes imágenes (aquellas que posteriormente son capturados por el robot). Cuando un conjunto de imágenes consecutivas son etiquetadas como no-clasificadas, un nuevo nodo tiene que ser creado con ellas y los representativos de los nodos tienen que ser recalculados.

4.3.3. Condición de prominencia

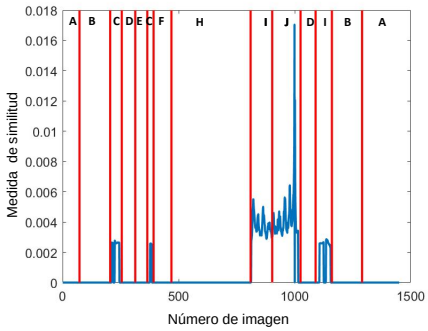
Como se ha explicado en la subsección 4.3.1, la etapa Node Level Loop Closure selecciona el conjunto N^* con los posibles nodos de pertenencia. De entre ellos, la etapa Image Level Loop Closure (subsección 4.3.2) selecciona la imagen I_i que cierra el bucle. El nodo al que pertenece I_i es también seleccionado como nodo de la nueva imagen I_q. Pero antes de seleccionar el nodo N_i como nodo de la nueva imagen, la imagen detectada como más similar a I_i debe cumplir la condición de prominencia.

La prominencia es evaluada en la curva de similitud (figura 4.3 (c)) y mide cuanto de pronunciado es el pico de la curva debido a su altura intrínseca y respecto a los valores vecinos. Esta condición es importante porque, no solo el valor de similitud tiene que ser alto para cerrar el ciclo, sino que este pico debe ser pronunciado respecto a los valores vecinos. La imagen seleccionada de la etapa Image Level Loop Clousure debe cumplir la condición presentada en la ecuación 4.6. En esta ecuación, P_{I_i} es el valor de prominencia del candidato, $\mu(P_{I_{k^*}})$ es la prominencia media de los candidatos contenidos en N^*. Por último, γ es un parámetro que determina cuanto de mayor debe ser la prominencia respecto a la media, cuyo valor se ha ajustado empíricamente a partir de los experimentos en un valor $\gamma = 5$.

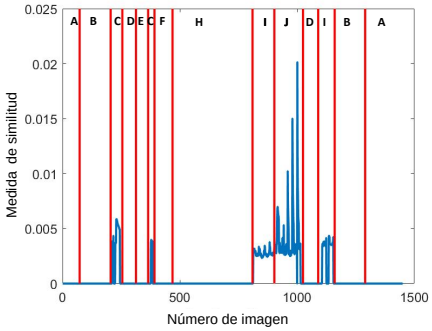
$$P_{I_i} \geq \gamma * \mu(P_{I_{k^*}}) \tag{4.6}$$

4.3.4. Condición de centroide

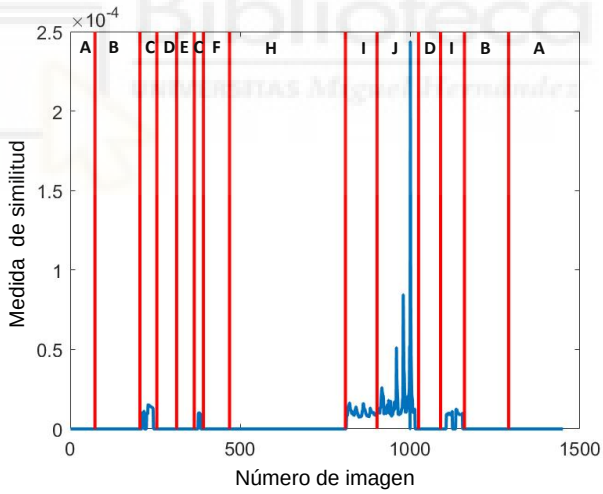
Una vez superada la condición de prominencia, y para que la imagen se incluya definitivamente en el nodo correspondiente se debe superar también la condición de centroide. Para que el nodo seleccionado sea aceptado como nodo final, se debe cumplir la ecuación 4.7. Esta condición estudia cuánto cambia la posición del representativo si se introduce la nueva imagen en el presente nodo. En esta condicion la distancia entre la posición del representativo antes de añadir la imagen (μ^{N_i}) y la posición tras añadir



(a) Similitud entre el descriptor de posicion de I_q y los descriptores de posición de las imágenes del conjunto de nodos candidatos.



(b) Similitud entre el descriptor de orientación de I_q y los descriptores de orientación de las imágenes del conjunto de nodos candidatos.



(c) Similitud final entre la I_q y las imágenes del conjunto de nodos candidatos.

Figura 4.3: Ejemplo de la etapa Image Level Loop Closure. La etiqueta de cada nodo está indicada en la parte superior de cada subfigura. En este ejemplo, la etapa Node Level Loop Closure da como nodos candidatos C, I y J.

la imagen($\mu^{N_i \cup \{\vec{d}_q\}}$) tiene que ser igual o menor a la influencia en el representativo efectuada por el resto de imágenes contenidas en el nodo.

$$\left|\vec{\mu}^{N_i} - \vec{\mu}^{N_i \cup \{\vec{d}_q\}}\right| \leq \max_j \left(\left|\vec{\mu}^{N_i - \{\vec{d}_j\}} - \vec{\mu}^{N_i}\right|\right) \tag{4.7}$$

donde $\vec{d}_j$ representa cada uno de los descriptores contenidos en el nodo N_i.

El proceso presentado entre las subsecciones 4.3.1 y 4.3.4 se puede resumir en el pseudocódigo Algorithm 1. El punto de inicio son los nodos calculados hasta el momento $N^C = \{N_1, N_2..., N_C\}$, su información asociada (N_l,$\vec{\mu}^{N_l}$ y $\sum_d^{N_l}$ para $l = 1, 2, ..., C$) y los descriptores de las imágenes previas $\vec{d}_k$. Con estos datos es posible determinar si una nueva imagen I_q, cuyo descriptor es $\vec{d}_q$, tiene que estar asignada o no a un nodo.

Algorithm 1 Pseudocódigo de las etapas Node e Image Level Loop Closure, aplicando las condiciones de Prominencia y Centroide.

NodeLevelLoopClosure ($\vec{d_q}$)
$N^C = \{N_1, N_2..., N_C\}$ conjunto inicial de nodos
$\vec{\mu}^{N_l}$, $\sum_d^{N_l}$ descriptor medio y covarianza del nodo N_l
for `l=1 hasta el numero de nodos C` **do**
    $\Delta_{\vec{d_q}}^{N_l} = (\vec{d_q} - \vec{\mu}^{N_l})^T (\sum_d^{N_l})^{-1} (\vec{d_q} - \vec{\mu}^{N_l})$ (ecuación (4.1))
    if $\left|\Delta_{\vec{d_q}}^{N_l} - \mu_{ns}^{N_l}\right| \leq \left|x\sigma_{ns}^{N_l}\right|$ **then** (ecuación (4.2))
        $N^* \leftarrow N_l$;
    end if
end for
end NodeLevelLoopClosure
ImageLevelLoopClosure ($\vec{d_q}$, $\vec{d_k}$, N^*)
for `k= 1 hasta imágenes en` N^* **do**
    $dist_{eucl}{}_{\vec{d_q}}^{\vec{d_k}} = \sqrt{\sum_{j=1}^{t} ((\vec{d_q}(j) - \vec{d_k}(j))^2}$ (ecuación (4.3));
    encontrar la imagen I_i más similar usando la ecuación (4.5);
    if I_i cumple $P_{I_i} \geq \gamma * \mu(P_{I_{k^*}})$ **then** (Condición de Prominencia ec. (4.6))
        if I_i cumple $\left|\vec{\mu}^{N_i} - \vec{\mu}^{N_i \cup \{\vec{d_q}\}}\right| \leq$ máx$_j$($\left|\vec{\mu}^{N_i - \{\vec{d_j}\}} - \vec{\mu}^{N_i}\right|$) **then** (Condición de Centroide ec.(4.7))
            I_i y N_i cierran el bucle;
            I_q se incluye en N_i;
            N_i, μ^{N_i} y $\sum_d^{N_i}$ se actualizan;
        else
            I_i no cumple la ec. ((4.7));
            I_q no se incluye en N_i;
        end if
    else
        I_i no cumple la ec. ((4.6));
        I_q no se incluye en N_i;
    end if
end for
if $N^* = \emptyset$ **then**
    I_q no se asigna a ningún nodo;
    los nodos actuales no son actualizados y I_q será evaluada en más adelante;
end if
end ImageLevelLoopClosure

4.3.5. Creación de nuevo nodo

El método presentado es capaz de detectar el nodo al que pertenece la nueva imagen I_q. Sin embargo, como se ha explicado anteriormente, hay ocasiones en las que la nueva imagen no se puede asignar a ningún nodo anterior. Esto puede ocurrir tanto si la nueva imagen es sustancialmente diferente a los descriptores representativos existentes o si la imagen elegida para cerrar el bucle no cumple con las condiciones de prominencia (subsección 4.3.3) o de centroide (subsección 4.3.4).

Cuando el número de imágenes consecutivas sin ser clasificadas es considerable, se crea un nuevo nodo N_q. El nuevo nodo tiene asociado un descriptor medio $\vec{\mu}^{N_q}$ con su correspondiente matriz de covarianzas $\sum {}_{d}^{N_q}$. Para modelar un nuevo nodo lo suficientemente diferente a los nodos ya creados en el mapa, la distancia media entre el descriptor representativo y los representativos de los nodos ya creados (utilizando la *distancia Euclídea*) tiene que ser mayor o igual a la distancia media entre los representativos que ya existen. Si el nuevo nodo creado cumple esta condición, N_q se añade al conjunto de nodos existentes y podrá ser seleccionado como posible nodo en la etapa Node Level Loop Closure.

Si el nuevo cluster no cumple esta condición, la etiqueta es eliminada y no podrá ser elegido en futuros procesos de la etapa Node Level Loop Closure. De este modo, la nueva imagen continúa siendo evaluada con los nodos que el mapa recoge hasta ese momento. Si la imagen no puede ser recogida en ninguno de los nodos presentes pero sí que es lo bastante similar a N_q, se incluye a este nuevo nodo. En el momento que se han unido las suficientes imágenes a N_q y que la distancia de su representativo al resto de representativos es mayor o igual que la distancia entre los representativos creados, el nodo N_q pasa a formar parte del mapa y pasa a ser elegible en la etapa Node Level Loop Closure. Esta condición intenta garantizar que los nodos que representan el entorno sean lo más diferente posible entre ellos y que no se extiendan excesivamente a lo largo el espacio de características.

4.3.6. Fusión de nodos

Siguiendo los procesos anteriores, el número de nodos aumenta constantemente conforme el robot avanza por el entorno. Pero puede darse el caso en el que el número resultante de grupos sea mayor de lo necesario y que imágenes que representan zonas adyacentes y/o visualmente similares pueden incluirse en un único nodo. El método que presentamos tiene en cuenta esta posibilidad y cuenta con la opción de disminuir el número de nodos si fuera necesario. Esto es posible porque el método detecta cuando dos nodos son suficientemente parecidos como para aglutinar la información en único nodo, dejando de esta manera un mapa más coherente tras la fusión.

Se introduce una nueva condición y un nodo N_q tiene que superar un umbral de disimilitud específico con los otros nodos. Si un nodo existente y uno nuevo son similares, se fusionan en un cluster único. Para resolver este problema, se utiliza la *distancia de Mahalanobis*, similar al proceso utilizado en la etapa Node Level Loop

Closure. La ecuación (4.8) muestra esta condición en el proceso de fusión de nodos

$$\Delta_{N_q}^{N_o} = (\vec{\mu}^{N_q} - \vec{\mu}^{N_o})^T (\sum {}_{d}^{N_o})^{-1} (\vec{\mu}^{N_q} - \vec{\mu}^{N_o}) \tag{4.8}$$

Para detectar si N_q debe unirse a un nodo N_o, se evalúa $\Delta_{N_q}^{N_o}$. Esta distancia tiene que satisfacer la condición de similitud presentada en la ecuación (4.9), donde $\mu_{ns}^{N_o}$ y $\sigma_{ns}^{N_o}$ son la media y desviación estandar de una distribución gaussiana calculada con la información del nodo N_o, calculada como se detalla en la subsección 4.3.1. y es un parámetro ajustable (cuadro 4.6). Cuanto más grande es el valor de y, menos restrictiva es la condición y el algoritmo es más propenso a fusionar nodos. Los experimentos muestran que este parámetro debe tomar un valor entre 1 y 2. Si $y = 1$ la condición es muy restrictiva, por tanto menos uniones de nodos se harán y más clusters habrá en el mapa final; pero cuando $y = 2$ la condición es menos restrictiva, de este modo se realizarán más fusiones y aparecerán menos nodos al final del experimento. Los experimentos para evaluar el valor de y son presentados en la sección 4.5.

$$\left|\Delta_{N_q}^{N_o} - \mu_{ns}^{N_o}\right| \leq \left|y\sigma_{ns}^{N_o}\right| \tag{4.9}$$

4.4. Conjunto de imágenes y base de datos

El método propuesto ha sido probado en diferentes entornos de interior utilizando conjuntos de imágenes capturadas bajo condiciones reales. Las bases de datos utilizadas han sido la base de datos INNOVA [5] capturada por el grupo de investigación ARVC al cual pertenezco y la base de datos COLD [135], capturada por terceros, disponible públicamente, que proporciona trayectorias de robots en algunos edificios de las universidades de Freiburg y Saarbrücken. Estas tres trayectorias brindan una buena opción para realizar mapeos incrementales de manera jerárquica ya que representan entornos reales y dinámicos que experimentan fenómenos típicos que pueden ocurrir durante la operación real como son el ruido, oclusion de un porcentaje de imagen, condiciones cambiantes de iluminación, movimiento de personas o incluso de recolocación algunos objetos o muebles. Por esa razón, estas imágenes constituyen un escenario desafiante para probar la solidez de las tareas de mapeo incremental.

Las secuencias de imágenes con las que se ha trabajado han sido capturadas por un robot móvil equipado con una cámara omnidireccional, formada por un espejo hiperbólico montado frontalmente en una cámara. El programa recibe las imágenes omnidireccionales y las transforma a imágenes panorámicas, una vez las imágenes son panorámicas estas pueden ser utilizadas en el proceso de mapeo. Adicionalmente, el robot está equipado con encoders en las ruedas y con láseres para estimar la posición. Estos sensores no son utilizados durante nuestros experimentos de mapeo visual, aunque con ellos se puede obtener el *ground truth* y utilizarlo para propositos de comparación.

La figura 4.4 muestra el robot utilizado para capturar la base de datos de INNOVA y el sistema de visión que emplea para capturar las imágenes. La información del robot utilizado para capturar las trayectorias de la base de datos COLD se puede

Figura 4.4: Robot móvil y su sistema de visión en la base de datos de INNOVA [5].

(a) Imagen de INNOVA

(b) Imagen de Saarbrücken

(c) Imagen de Freiburg

Figura 4.5: Imágenes panorámicas de cada una de las bases de datos.

Cuadro 4.1: Distancia cubierta [m] y número de imágenes usadas en cada base de datos.

Trayectoria	Número de Imágenes	Distancia Cubierta
Ruta INNOVA	1450	176.26 m
Ruta Saarbrücken	1021	56.64 m
Ruta Freiburg	2778	102.68 m

encontrar en la referencia [135]. En la figura 4.5 se puede ver una muestra de imágenes capturadas en estos entornos.

Por último, el cuadro 4.1 muestras las especificaciones de las 3 trayectorias y el número de imágenes tomadas en cada trayectoria. Todas las rutas contienen cierres de bucle para poder testear las etapas de Nodo e Image Level Loop Closure.

Finalmente, consideramos esencial decir que estos entornos pueden ser problemanticos ya que a lo largo de la ruta hay muchas ventanas y paredes de vidrio y por tanto el clima exterior y las condiciones de iluminación podrían tener un impacto negativo en la tarea de mapeo. De este modo, se trata de entornos dinámicos y desafiantes, condiciones que crean un excelente banco de pruebas para nuestros experimentos.

4.5. Experimentos

En esta sección se presentan los resultados obtenidos tras la ejecución del método propuesto para la creación de mapas jerárquicos de manera incremental utilizando las bases de datos presentadas en la sección anterior. Primero, un conjunto de imágenes son tomadas para crear el primer nodo en un método supervisado. Una vez el primer nodo ha sido creado, el proceso comienza y crea el mapa mientras el robot se desplaza por el entorno, actualizando la información del mapa cada vez que una nueva imagen es tomada.

4.5.1. Parámetros elegidos para describir las imágenes

Como se ha explicado en la sección 4.2, al utilizar los descriptores de apariencia global, cada imagen panorámica es reducida a un vector $\vec{d} \in \mathbb{R}^{l \times 1}$ cuyo tamaño dependerá de los parámetros utilizados durante el proceso de descripción. Estos parámetros están reflejados en el cuadro 4.4. En este trabajo, se utiliza $\vec{d_h} \in \mathbb{R}^{256 \times 1}$ cuando se calcula el descriptor de posición HOG. Para el decriptor de orientación, $b_{ho} = 16$, $k_{ho} = 256$ y $dist_{ho} = 2$, de este modo la imagen panorámica es reducida a un vector de tamaño $\vec{d_v} \in \mathbb{R}^{4096 \times 1}$. Utilizando el descriptor gist, los parámetros se mantienen en $m_{gp} = k_{gp} = 16$ y $r_{gp} = 2$ para el descriptor de posición, teniendo así un vector $\vec{d_p} \in \mathbb{R}^{512 \times 1}$. Por último, cuando se requiera calcular el descriptor gist para orientación, este es calculado con $m_{go} = 16$, $k_{go} = 256$, $dist_{go} = 2$ y $r_{go} = 1$, transformando cada imagen en un vector $\vec{d_o} \in \mathbb{R}^{4096 \times 1}$.

4.5.2. Parámetros para el proceso de cierre de bucle

Recordando el proceso utilizado para la creación de mapas detallado en la sección 4.3, existen unos párametros de los cuales dependerán los resultados obtenidos tras la tarea de mapeo. En esta subsección se explicarán los valores de parámetros elegidos.

Cuando una nueva imagen es capturada, el primer paso es conocer si pertenece a alguno de los nodos creados. Como se detalla en la subsección 4.3.1, para llevar a cabo el cálculo de la etapa Node Level Loop Closure se utiliza la *distancia de Mahalanobis* (ecuación (4.1)). Después de este cálculo es posible determinar si un descriptor $\vec{d}_q$ puede pertenecer a un nodo específico N_i y ser almacenado en el conjunto de posibles nodos N^* utilizando la ecuación (4.2).

Una vez los nodos candidatos son seleccionados, sus imágenes se comparan con el nuevo descriptor utilizando la *distancia Euclidea* (ecuación (4.3)), donde d^{N_k} son todos los descriptores candidatos que provienen de la etapa de cierre de nodo. La etapa Image Level Loop Closure detecta la imágen más similar a la nueva imagen adquirida y la nombra I_i. Después de detectar I_i, esta debe satisfacer las condiciones de Prominencia y Centroide (subsecciones 4.3.3 y 4.3.4). Por último, cada vez que el contenido de un nodo cambia, se compara con el resto de nodos por si es posible realizar una fusión de nodos (subsección 4.3.6).

Como se ha descrito, estas ecuaciones dependen de parámetros ajustables con influencia en el resultado de mapeo. Estos parámetros deben ser adaptados a la aplicación que se quiera realizar y/o para los resultados que se quiera esperar. La ecuación (4.2) depende del valor de x. Este valor determina cuanto de restrictiva es la etapa de Node Level Loop Closure; cuanto menor es x, mayor es la facilidad para cerrar el bucle con los nodos ya creados. El parámetro x se ha determinado que sea variale respecto al numero de nodos creados (C) y siguiendo la expresión $x = 3{,}05 - 0{,}15 * C$. El valor de x se irá reduciendo cuando la cantidad de nodos creados aumente, pero este valor se verá limitado por Ω, la figura 4.2 muestra gráficamente estos valores. Ω será uno de los parámetros que variaremos en los experimentos para determinar su influencia en el proceso. Otro parámetro variable es y, que es utilizado en la ecuación 4.9 y de su valor depende la cantidad de nodos fusionados; cuanto menor es el valor de y, más facil es fusionar nodos similares. Finalmente se encuentra el parámetro γ, que aparece en la ecuación (4.6). Una vez I_i es recuperada, esta ecuación evalúa la prominencia del resultado para garantizar que la imagen elegida para cerrar el bucle tiene un pico en la curva de similitud lo suficientemente alto en comparación con sus vecinos. Cuanto mayor es el valor γ, más prominente debe ser el pico de I_i en la curva de similitud. Durante los experimentos cuyos resultados se han expuesto en esta sección γ mantiene un valor constante, $\gamma = 5$. Todos estos parámetros están expuestos en el cuadro 4.6.

4.5.3. Evaluación

Una de las formas de evaluar la destreza del método descrito durante este capítulo es hacer uso del *ground truth* y comparar si visualmente los mapas han sido

clusterizados de la mejor manera posible. Esta forma de evaluación sería subjetiva y no daría una valor cualitativo para caracterizar los resultados. Surge por tanto la necesidad de realizar una evaluación con algún valor que muestre como de buena ha sido la separación en nodos. Con esta idea se presenta el valor *Silhouette* (valor de silueta). La silueta evalua la compacidad de la clusterización; calcula el grado de similitud entre un descriptor y los descriptores que pertenecen al mismo nodo y lo compara con los descriptores pertenecientes a otros nodos. El valor medio de silueta (S) de todas las entidades se calcula utilizando la ecuación (4.10). Cuanto mayor es el valor S significa que más similares son los descriptores de un mismos grupos y más distintos son a descriptores de otros nodos. El máximo valor de silueta es 1 que indica que ha habido una buena separación en la clusterización, dando lugar a nodos con entidades muy similares entre si y muy diferentes a las entidades de otros nodos. Por el contrario, el valor mínimo es -1, que significa que los clusters resultantes no separan correctamente la información. En este trabajo, se utilizará este valor de silueta para evaluar la compacidad de los custers, esta es calculada usando los puntos de captura de las imágenes.

$$S = \frac{\sum_{i=1}^{C} s_i}{C} \tag{4.10}$$

En la ecuación (4.10), C es el número de nodos y s_i la silueta media de los descriptores contenidos en el nodo i. La silueta de cada descriptor $\vec{d_j}$ se calcula usando la ecuación (4.11).

$$s_j = \frac{b_j - a_j}{max(a_j, b_j)} \tag{4.11}$$

donde a_j es la distancia media entre el punto de captura de $\vec{d_j}$ y el punto de captura de los otros descriptores contenidos en el mismo nodo y b_j es la distancia media entre el punto de captura de $\vec{d_j}$ y el punto de captura de las imágenes clusterizadas en otros nodos. La información sobre los puntos de captura se conoce porque está disponible el *ground truth* con los datos de adquisición. No obstante, cabe destacar que la tarea de mapping se realiza únicamente con información visual y el *ground truth* se utiliza solamente para cuantificar el rendimiento de la tarea de mapeo incremental.

Sin embargo, el valor de silueta debe ser un valor orientativo pero no tiene por que determinar, en valores absolutos, si un resultado es mejor. Dependiendo de la aplicación futura puede ser necesario crear un menor número de nodos, aunque existan distribuciones más optimas con otro número de nodos. De tal manera, que es más aconsejable comparar los valores de silueta en situaciones de mismo número de nodos.

Adicionalmente al valor de silueta, el número de nodos obtenidos durante el proceso también se mostrará en la subsección de resultados. De esta manera se muestra y evalúa como los parámetros tienen influencia en el número final de nodos. De este modo, es posible evaluar los resultados y seleccionar la mejor configuración por compacidad de los nodos creados o por la cantidad de estos.

4.5.4. Resultados

4.5.4.1. Influencia de los parámetros en los resultados del algoritmo

La figura 4.6 muestra los resultados obtenidos tras realizar el mapping incremental utilizando HOG como descriptor de las imágenes y la figura 4.7 cuando se utiliza gist como descriptor. En ambas, se muestran los resultados de las diferentes bases de datos: (a) INNOVA, (b) Saarbrücken y (c) Freiburg. En ambas figuras, es posible visualizar dos tipos diferentes de resultados. En una primera fila se muestra el número final de nodos obtenidos tras realizar el proceso de mapeo con todas las imágenes de la trayectoria, en la segunda fila se muestra el valor medio de silueta. Estas figuras muestran la influencia de los parámetros y y Ω, parámetros introducidos en el cuadro 4.6.

Primero, evaluando el parámetro y, como se esperaba, cuanto mayor es el valor de y menor es el número de nodos al evaluar toda la trayectoria. El efecto de este parámetro es menos representativo al utilizar el descriptor de gist, tal y como se puede ver en la figura 4.7. Por lo que respecta a la variación del parámetro Ω, el cual limita x en la ecuación (4.2); si su valor es bajo, los nodos existentes lo tendrán más difícil para superar la etapa de Node Level Loop Closure y será más fácil encontrar imágenes que en un primer momento no son añadidas a ningún nodo. Esto puede provocar una mayor facilidad a que imágenes consecutivas son determinadas en un primer momento que no pertenecen a ningún nodo promoviendo la creación de nuevos nodos. Sin embargo, si el valor de Ω es alto (limitando a x a tener valores altos), más nodos pueden cumplir la etapa de Node Level Loop Closure dificultando en parte la obtención de imágenes consecutivas consideradas de no pertenencia a ninguno de los nodos existentes. Como muestran las figuras 4.6 y 4.7, cuando se usa HOG, el parámetro y tiene una mayor influencia que Ω en el número final de nodos, mientras que utilizando el descriptor gist, este último muestra una mayor importancia en los resultados.

La segunda fila de cada entorno en las figuras 4.6 y 4.7 muestra la silueta media despues del proceso de mapping. En general, valores medios de Ω ofrecen valores de silueta más altos mientras que el efecto de y es más variable. Como recordatorio, cuanto mayor es el valor de silueta, mejor distribuidas están las imágenes en los nodos. Aunque, como se ha visto en el subapartado anterior, el valor de silueta debe ser un valor orientativo sobre la compacidad del resultado final. Nos puede dar una idea de cuánto de bueno es, pero no debe ser tratada como característica absoluta. Es más aconsejable comparar los valores de silueta en situaciones en las que se ha obtenido el mismo número de nodos.

Para un mejor estudio de los resultados, el cuadro 4.2 muestra los máximos valores de silueta en el resultado y el número de nodos con los que se ha obtenido dicha configuaración. Además, también muestra el valor de los parámetros que conducen a esa silueta máxima. Este cuadro muestra que valores bajos de Ω, como valores cercanos a 1.85, ofrecen los mejores resultados, aunque cuando se usa el descriptor *gist* también se obtienen buenas valores de silueta con valres de Ω altos. Por otro lado, los resultados no dependen sustancialmente del valores de y. Como se puede observar,

Cuadro 4.2: Silueta máxima obtenida en cada ruta, mostrando el numero de nodos y la configuración de parámetros.

HOG			
Trayectoria	**Silueta**	**Nodos**	**Valor de parámetros**
Ruta INNOVA	0.3973	6	$y = 2{,}25$, $\Omega = 1{,}7$
Ruta Saarbrücken	0.2756	16	$y = 1{,}25$, $\Omega = 1{,}85$
Ruta Freiburg	-0.1526	30	$y = 0{,}75$, $\Omega = 1{,}7$
Gist			
Trayectoria	**Silueta**	**Nodos**	**Valor de parámetros**
Ruta INNOVA	0.2556	6	$y = 1{,}5$, $\Omega = 2{,}15$
Ruta Saarbrücken	0.1262	19	$y = 1{,}5$, $\Omega = 1{,}85$
Ruta Freiburg	-0.1449	34	$y = 1{,}5$, $\Omega = 2{,}3$

se obtienen mejores resultados utilizando el descriptor HOG. Este comportamiento ha sido observado en diferentes experimentos y situaciones, lo que hace concluir que HOG es una opción más adecuada para describir las imágenes con el objetivo de crear un mapa de forma incremental. Cabe mencionar que la silueta obtenida en la base de datos de Friburg es sustancialmente más baja que los resultados obtenido con las otras dos bases de datos. La razón puede ser doble. Por un lado, la ruta de Friburgo es más larga, obteniendo grandes espacios que son visulmente muy parecidos, esto puede influir bastante en el algoritmo de mapeo. Por otro lado, el entorno de Friburgo contiene varias paredes transparentes, y numerosas ventanas de cristal, esto puede producir saturación en las imágenes y mezcla entre la información de las habitaciones adyacentes. Estos efectos pueden provocar alteraciones a la hora de describir globalmente las imágenes, así como tener un impacto negativo sobre el desempeño de la tarea de mapping.

4.5.4.2. Resultados con clustering espectral absoluto

Con fines comparativos, se estudia el resultado de un clustering espectral [33] como punto de referencia con el que contrastar los resultados. Cabe destacar que este algoritmo de clustering espectral es absoluto, es decir, tiene información completa sobre todos los descriptores desde el inicio del proceso, pudiendo calcular todas las similitudes mutuas para realizar el proceso de agrupamiento de forma off-line. Por tanto, constituye un punto de referencia muy potente con el que comparar el rendimiento relativo de nuestra propuesta. Para utilizar esta técnica, el número de nodos está fijado previamente y todas las imágenes están disponibles desde un inicio, por esta razón esta técnica no es una opción adecuada para contruir mapas de manera incremental. Las figuras 4.8 y 4.9 muestran los resultados tras utilizar el método de clustering espectral absoluto y los descriptores HOG y gist respectivamente. Estas figuras muestran la distancia media respeto a un número de nodos preestablecido.

La figura 4.8 muestra la silueta media cuando se utiliza el descriptor HOG. La figura muestra como la silueta media decrece cuando el número de nodos preestablecidos previamente aumenta. Los resultados obtenidos en INNOVA muestran que la silueta

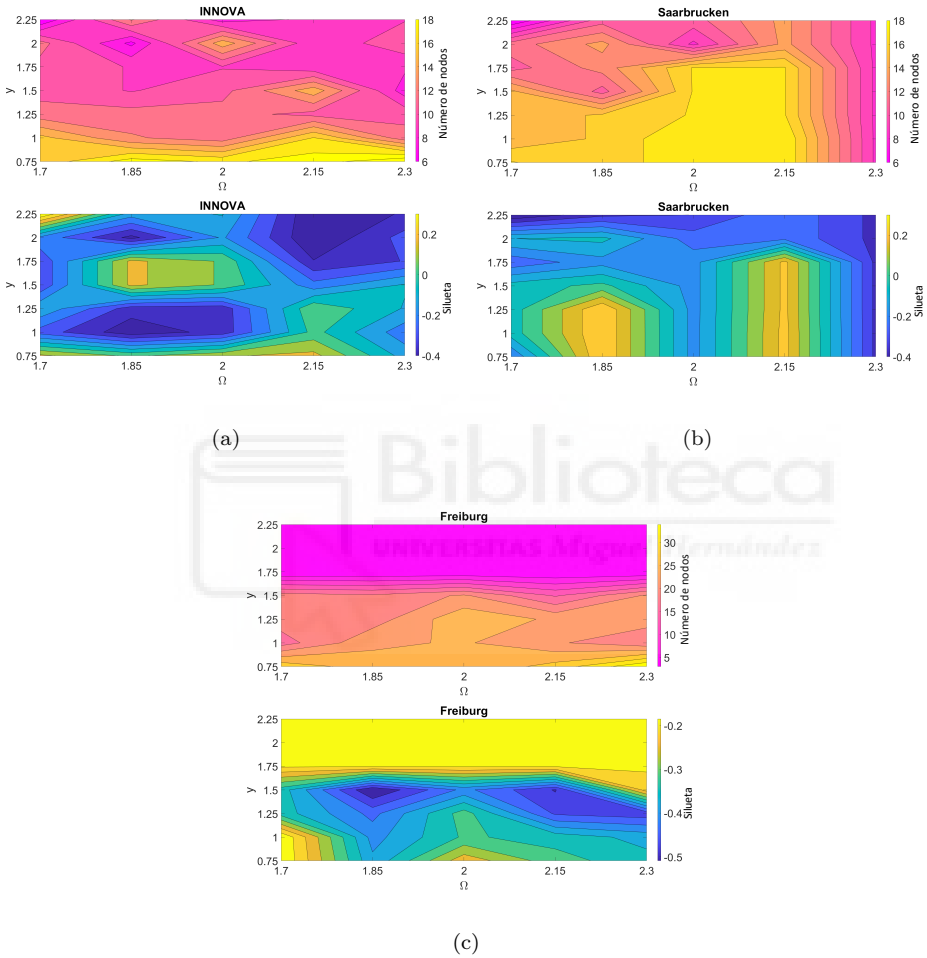


Figura 4.6: Resultados obtenidos con el descriptor HOG para diferentes valores de y y Ω para las bases de datos (a) INNOVA, (b) Saarbrücken y (c) Freiburg. El primer mapa de calor de cada subfigura muestra el **número final de nodos** y el segundo la **silueta** media después de realizar el mapping incremental.

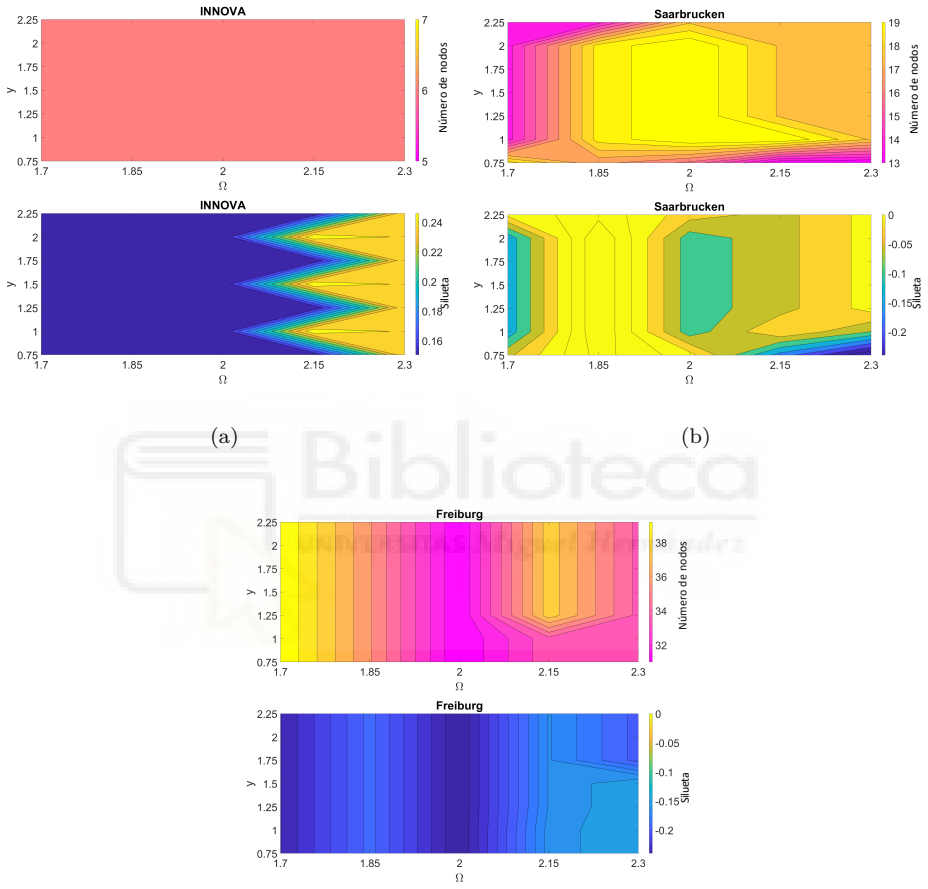


Figura 4.7: Resultados obtenidos con el descriptor gist para diferentes valores de y y Ω para las bases de datos (a) INNOVA, (b) Saarbrücken y (c) Freiburg. El primer mapa de calor de cada subfigura muestra el **número final de nodos** y el segundo la **silueta** media después de realizar el mapping incremental.

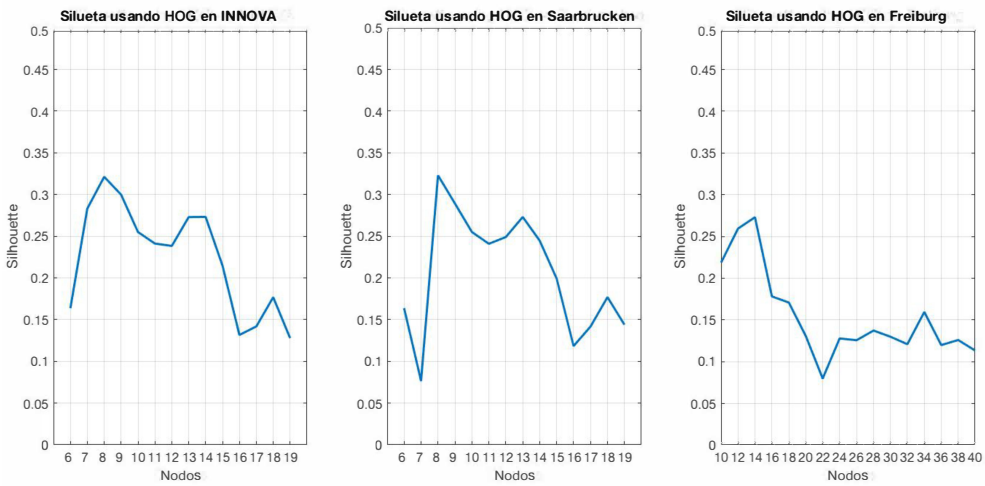


Figura 4.8: Silueta media tras utilizar clustering espectral y HOG versus número de nodos.

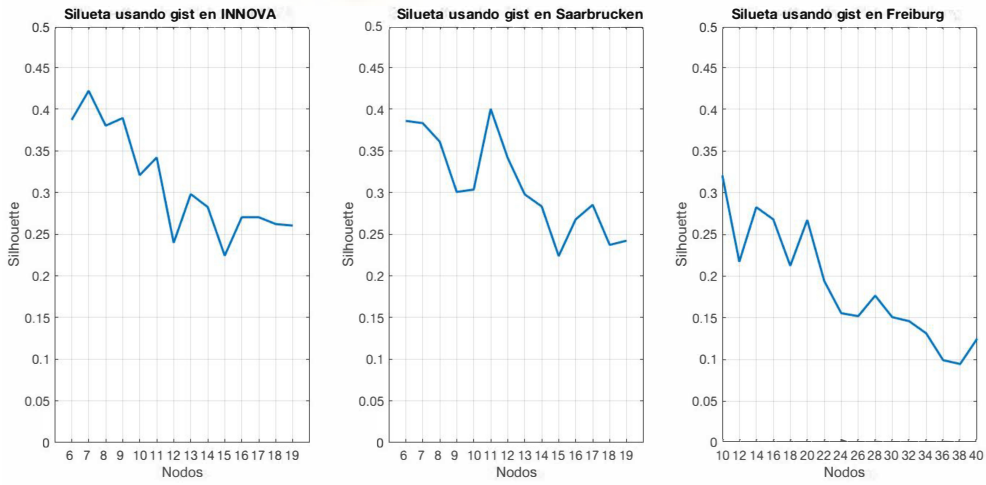


Figura 4.9: Silueta media tras utilizar clustering espectral y gist versus número de nodos.

puede situarse en valores entre 0.1 y 0.3; en particular, si nos fijamos en el resultado obtenido con 6 nodos (número de nodos con el que se había obtenido la mejor silueta con nuestro método de clustering incremental) se obtiene una silueta de 0.13 mientras que con nuestro método la silueta obtenida es de 0.3973. En el caso de la ruta de Saarbrücken, las siluetas obtenidas con el método de clustering espectral están entre 0.1 y 0.3 y los resultados con el método propuesto entre -0.4 y 0.3. Observando el resultado cuando se establecen 16 nodos (con el que obteníamos nuestro mejor dato de silueta) el valor de silueta es cercano a 0.15 mientras que con el método propuesto es de 0.2756. Finalmente, teniendo en cuenta la base de datos de Freiburg, los resultados utilizando el clustering espectral están entre 0.15 y 0.25 como valor de silueta, mientras que utilizando el método de clustering incremental los valores están entre -0.5 y -0.15. El mejor resultado con nuestro método se obtiene con 30 nodos y un valor de silueta de -0.1526, con la referencia de clustering espectral evaluada con este número de nodos se obtiene un valor de silueta de 0.13.

Adicionalmente la figura 4.9 muestra la silueta obtenida al utilizar el descriptor gist. En este caso los resultados son menos similares a los resultados obtenidos con el método propuesto de clustering incremental. Como antes, cuanto mayor es el número de nodos a calcular, menor es el valor de silueta obtenido con el método de clustering espectral. La mejor silueta obtenida con gist en INNOVA utilizado el método propuesto en este capítulo es 0.2556 con 6 nodos. Utilizado el método de clustering espectral este valor es de 0.38 para el mismo número de nodos. Con Saarbrücken la silueta obtenida está entre 0.25 y 0.4 y el mejor valor que obteniamos con nuestro método es obtenido con 19 nodos y un valor de silueta 0.1262. Finalmente, el resultado con la base de datos de Freiburg empleando el clustering espectral está entre 0.1 y 0.25, pero emplenado el proceso de clustering incremental el mejor resultado es -0.1449 obtenido con 34 nodos. Para 34 nodos en Freiburg y el método de clustering espectral la silueta es de 0.15

El resultado de la evaluación entre el método de clustering incremental y jeráquico y el punto de referencia ofrecido por el clustering espectral absoluto queda resumido en el cuadro 4.3. Este cuadro incluye la silueta máxima obtenida por el método propuesto y la silueta obtenida con el clustering espectral para el mismo número de nodos. Es necesario resaltar que para el método propuesto se han obtenido mejores resultados utilizando HOG en las rutas de INNOVA y Saarbrücken. Este hecho es especialmente relevante ya que cabe recordar que el método de clustering incremental explicado permite contruir el mapa mientras el robot explora el entorno, trabajando con información visual incompleta. Sin embargo, el método de clustering espectral necesita toda las imágenes capturadas antes de iniciar el algoritmo. Los resultados con la base de datos de Freiburg son menos concluyentes, debido a las condiciones especiales comentadas sobre esta ruta y a su mayor extensión. HOG se destaca nuevamente como un método de descripción de imágenes eficiente para tareas de mapping incremental.

4.5.4.3. Vista en planta de los puntos de captura

Las figuras 4.10–4.12 muestran la vista en planta de los puntos de captura de las tres rutas al final de la exploración del entorno y la construcción del mapa de forma

Cuadro 4.3: Comparación entre la máxima silueta obtenida con clustering incremental y la silueta obtenida con clustering espectral para el mismo número de nodos.

HOG			
Trayectoria	**Nodos**	**Clustering incremental**	**Clustering espectral**
Ruta INNOVA	6	0.3973	0.13
Ruta Saarbrücken	16	0.2756	0.15
Ruta Freiburg	30	-0.1526	0.13
Gist			
Trayectoria	**Nodos**	**Clustering incremental**	**Clustering espectral**
Ruta INNOVA	6	0.2556	0.38
Ruta Saarbrücken	19	0.1262	0.25
Ruta Freiburg	34	-0.1449	0.15

incremental. Los puntos de captura se muestran con diferentes formas y colores, dependiendo del nodo al que pertenecen. Además, las figuras 4.13–4.15 muestran el mapa de las rutas en diferentes puntos del proceso de construcción del mapa incremental.

Si se presta atención a las diferentes subfiguras, se puede observar como los nodos se van creando mientras el mapa se actualiza. Si se observan las subfiguras 4.13c,d,f,g o 4.15e,f es posible ver el proceso de fusión de nodos (Section 4.3.6). Inicialmente hay diferentes nodos y tras incluir nuevas imágenes y crear o modificar los nodos, el proceso contempla la posibilidad de fusionar nodos debido a su similitud. Como resultado se obtiene el mapa fragmentado en un menor número de nodos debido a la fusión de estos. Además, es posible comprobar que el hecho de tener más imágenes no tiene como resultado un mayor número de nodos, es el hecho de pasar por un mayor número de estancias. Por ejemplo en la subfigura 4.14b, tras 399 imágenes el algoritmo ha detectado 7 nodos. En la subfigura 4.14e con 785 imágenes se han detectado 14 nodos distintos y desde ese punto hasta el final del recorrido el robot solo se desplaza por un largo pasillo y cambia una única vez de estancia, como resultado solo se crean 2 nuevos nodos desde ese punto hasta que la exploración finaliza. Otras características como cierres de bucle se pueden observar en las figuras.

Figura 4.10: Mapa obtenido utilizando HOG, $y = 2{,}25$ y $\Omega = 1{,}7$ en la base de datos de INNOVA. El proceso detecta 6 nodos y un valor medio de silueta de 0.3973.

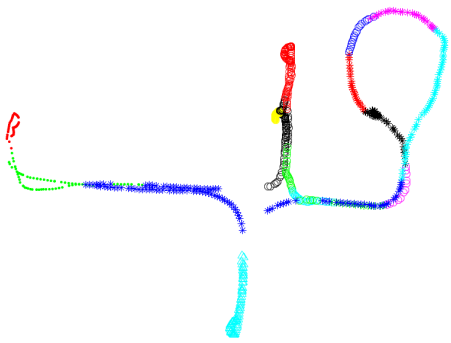


Figura 4.11: Mapa obtenido utilizando HOG, $y = 1{,}25$ y $\Omega = 1{,}85$ en la ruta de Saarbrücken dataset. El proceso detecta 16 nodos y un valor medio de silueta de 0.2756.

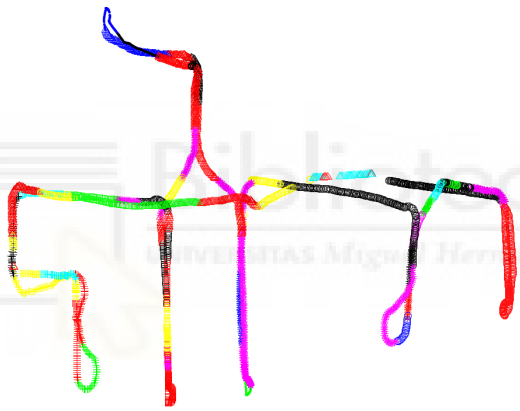
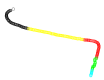


Figura 4.12: Mapa obtenido utilizando HOG, $y = 0{,}75$ y $\Omega = 1{,}7$ en la ruta de Freiburg. El proceso detecta 30 nodos y un valor medio de silueta de -0.1526.

(a) 240 imágenes y 3 nodos.



(b) 286 imágenes y 5 nodos.

(c) 344 imágenes y 5 nodos.

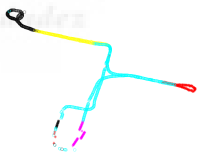
(d) 417 imágenes y 4 nodos.

(e) 537 imágenes y 8 nodos.

(f) 640 imágenes y 10 nodos.

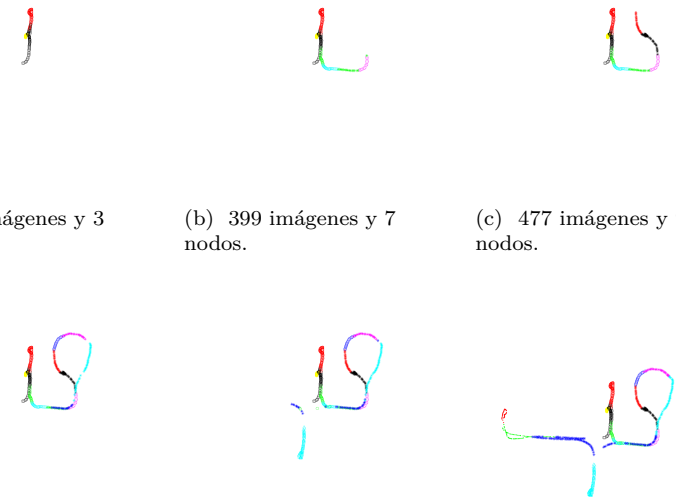
(g) 818 imágenes y 11 nodos.

(h) 985 imágenes y 6 nodos.



(i) 1450 imágenes y 6 nodos.

Figura 4.13: Mapa obtenido utilizando HOG, $y = 2{,}25$ y $\Omega = 1{,}7$ en la base de datos de INNOVA. Las subfiguras muestran algunos pasos intermedios del proceso. Los pares de subfiguras 4.13c-d y 4.13f-g muestran el proceso de fusión de nodos.



(a) 268 imágenes y 3 nodos.

(b) 399 imágenes y 7 nodos.

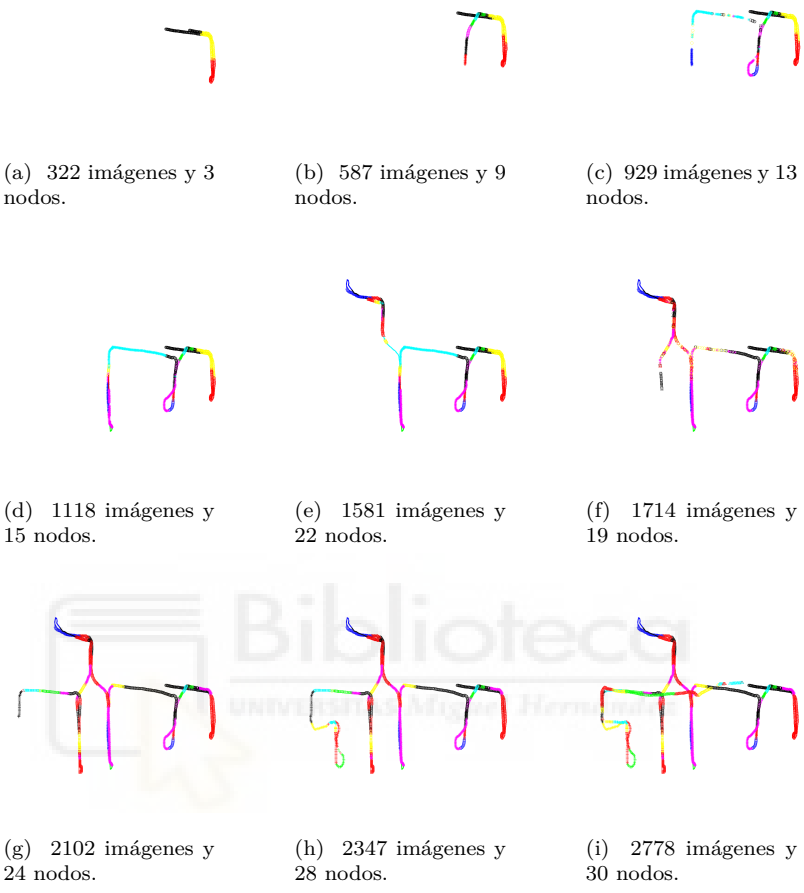
(c) 477 imágenes y 9 nodos.

(d) 629 imágenes y 13 nodos.

(e) 785 imágenes y 14 nodos.

(f) 1021 imágenes y 16 nodos.

Figura 4.14: Mapa obtenido utilizando HOG, $y = 1{,}25$ y $\Omega = 1{,}85$ en la ruta de Saarbrücken dataset. Las subfiguras muestran algunos pasos intermedios del proceso.



(a) 322 imágenes y 3 nodos.

(b) 587 imágenes y 9 nodos.

(c) 929 imágenes y 13 nodos.

(d) 1118 imágenes y 15 nodos.

(e) 1581 imágenes y 22 nodos.

(f) 1714 imágenes y 19 nodos.

(g) 2102 imágenes y 24 nodos.

(h) 2347 imágenes y 28 nodos.

(i) 2778 imágenes y 30 nodos.

Figura 4.15: Mapa obtenido utilizando HOG, $y = 0{,}75$ y $\Omega = 1{,}7$ en la ruta de Freiburg. Las subfiguras muestran algunos pasos intermedios del proceso. El par de subfiguras 4.15e,f muestran el proceso de fusión de nodos.

4.6. Cambios y mejoras

Tras la publicación de los trabajos relativos a la confección de mapas incrementales de interior usando información visual y descriptores de apariencia global se lleva a cabo un estudio del algoritmo para detectar puntos débiles del programa y realizar mejoras. Tras la etapa de revisión se constata que el algoritmo implementado hasta ahora funciona correctamente en entornos acotados y con un máximo aproximado de en torno a 15 nodos, pero su buen funcionamiento se ve degradado cuando el tamaño del entorno aumenta o cuando se revisitan escenarios, pero la información de estos ha variado visualmente con efectos adversos como podrían ser cambios en la iluminación. Por ello se abre una etapa de mejoras y cambios en el algoritmo presentado en este capítulo. Durante esta nueva etapa de mejoras de esta aplicación se llega a diferentes

modificaciones que hacen que el algoritmo sea más robusto ante entornos cambiantes y/o extensos. A continuación, se describe los principales cambios que se proponen.

Por un lado, la etapa *Node Level Loop Closure* mantiene el mismo funcionamiento que en la primera versión. Esta etapa trata de detectar los posibles nodos a los que la nueva imagen puede pertenecer, los nodos que cumplen la ecuación (4.2) se consideran candidatos a cierre de ciclo y se introducen en el grupo N^*.

A continuación, se activa la etapa *Image Level Loop Closure*. En esta etapa se trata de encontrar las imágenes con mayor similitud visual con la nueva imagen I_q para poder determinar de esta manera que se ha cerrado el ciclo y se puede añadir la nueva imagen a un nodo previo elegido como cierre de ciclo. Esta etapa ha sufrido unas modificaciones respecto al funcionamiento de la primera versión. Anteriormente la idea principal era encontrar la imagen I_i más similar a I_q. Con ello, el nodo N_i al que pertenece I_i era considerado como nodo de cierre de ciclo. Este proceso se modifica para tener en cuenta lo siguiente; no solo se trata de encontrar la imagen más similar de entre las imágenes de los nodos elegidos como posibles, sino que se trata de encontrar un entorno en la que la nueva imagen sea bastante similar a las imágenes del nodo con el cual va a cerrar el ciclo.

Siguiendo este concepto, el inicio de esta etapa es el mismo que el descrito hasta ahora. En primer lugar, se obtiene información de posición y orientación utilizando los descriptores globales y la distancia *Euclidean* (ecuación (4.3)). Una vez obtenida la distancia posición y la distancia orientación se calcula lo que se ha denominado similitud entre imágenes, es decir, se realiza la inversa de las distancias obteniendo valores altos para aquellas imágenes con una baja distancia entre descriptores y valores bajos para aquellas parejas con valores de distancia elevados. Tras normalizar los resultados, los valores de similitud posición y orientación se multiplican y se obtiene el valor similitud total (figura 4.16). En este punto el proceso empieza a ser diferente, el nodo N_i elegido como más similar es aquel en el que el valor similitud media es mayor, de este modo es más seguro determinar que en el nodo elegido la nueva imagen I_i es más similar al conjunto de imágenes pertenecientes al nodo. Para entender mejor el cambio es necesario hacer uso de la figura 4.16. Con la ejecución de la etapa *Image Level Loop Closure* como se había realizado hasta ahora se determinaría que el nodo más similar es el D, ya que en él se encuentra el valor de similitud más alto. Tras realizar nuevos experimentos, se determina que es mejor elegir el nodo en el cual la media de las distancias de similitud se parezca más a la nueva imagen, por tanto, se seleccionará B como el nodo más similar a la nueva imagen.

Como consecuencia de este cambio, la *condición de prominencia* no se calcula. El principal aliciente que suponía comprobar esta condición era asegurar que la similitud de la imagen elegida debía destacar mucho respecto a las vecinas. Con el cambio de paradigma en el que la imagen debe parecerse al conjunto de imágenes del nodo, esta condición pierde su valor y se elimina del algoritmo.

Siguiendo el proceso, el algoritmo es capaz de detectar a qué nodo pertenece la imagen I_q. Sin embargo, cabe la posibilidad de que la nueva imagen no sea asignada a

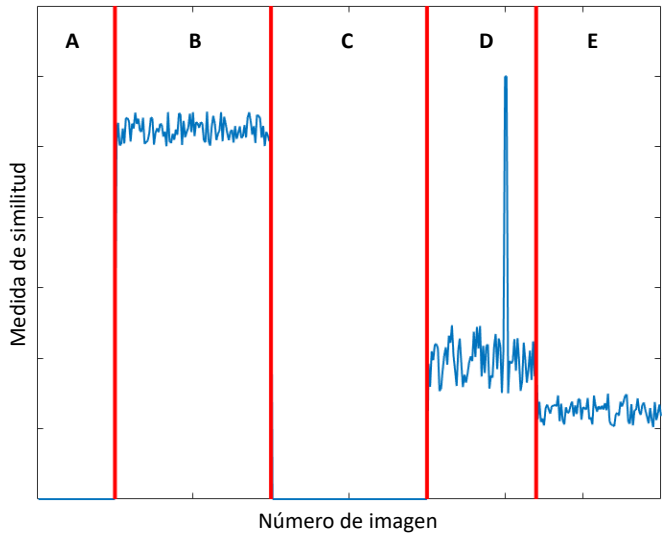


Figura 4.16: Valor de similitud total para la explicación de las mejoras en el algoritmo.

un nodo pretérito. Esto puede deberse a que ningún nodo haya superado la fase *Node Level Loop Closure* y por tanto no sea similar a ninguno de los nodos creados o porque la imagen no haya superado la condición de centroide (sección 4.3.4). La continua no asignación a uno de los nodos creados puede ayudarnos a concretar cuándo es necesario *crear un nuevo nodo*. Cuando un número considerable de imágenes consecutivas no son asignadas a ningún cluster, es el momento de crear uno nuevo. Tras realizar nuevos experimentos con bases de datos más extensas o con la revisita de lugares conocidos con nuevas trayectorias se ha detectado la necesidad de modificar las condiciones para crear un nuevo cluster. Por ello, esta condición de creación de nuevos nodos es diferente respecto de la primera versión del algoritmo.

Un nuevo nodo es creado cuando, de las últimas 20 imágenes, al menos 15 son designadas como imágenes sin nodo asignado. Si esta condición se cumple, se realiza una comprobación para detectar que el nuevo nodo sea suficientemente diferente a los nodos creados. Por otro lado, si de las últimas 30 imágenes, 28 son imágenes no asignadas, el nuevo nodo se crea obligatoriamente. Por último, otra condición para crear un nuevo nodo es cuando en las últimas 20 imágenes hay 9 cambios de nodo. Se presupone entonces que no hay un potencial nuevo nodo en esa región y habría que crearlo; para esta condición también se comprueba que el nuevo nodo sea suficientemente diferente a los nodos creados. Estas condiciones se pueden observar en la figura 4.17.

Para comprobar que un nuevo nodo es lo suficientemente diferente a los nodos creados se realiza una comprobación de distancia entre los representativos. Primero se calcula la distancia media entre el descriptor medio representativo del nuevo nodo y los descriptores representativos de los diferentes nodos ya creados. A continuación, se calculan las distancias entre los representativos de los nodos ya creados. Para aceptar

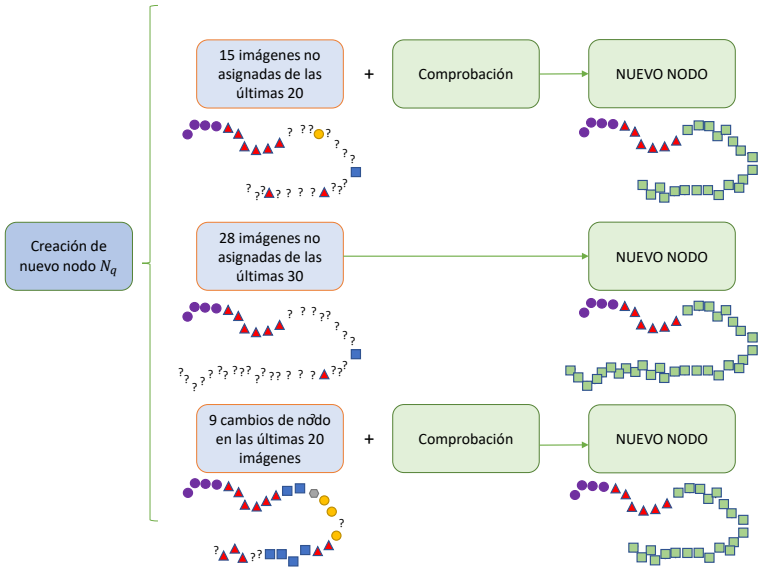


Figura 4.17: Cambios en la decisión de crear nuevos nodos.

el nuevo nodo, la distancia entre este y los nodos creados debe ser mayor a la distancia entre los nodos ya creados. Si hay que comprobar esta condición y no se cumple el nuevo nodo se elimina, no puede ser elegible en la fase *Node Level Loop Closure* y sus imágenes vuelven a considerarse como no asignadas a un nodo.

Si el nodo sí que puede ser creado, este nodo empieza a formar parte del conjunto de nodos y las imágenes que lo forman son reevaluadas. Adicionalmente, una ventana de imágenes previas también son reevaluadas para comprobar si continúan formando parte del nodo al que pertenecían previamente o son más similares al nuevo nodo. Para concluir la etapa, todos los representativos de los nodos son recalculados.

4.6.1. Presentación de resultados

Tras realizar estos cambios se presentan las figuras con el algoritmo mejorado. A continuación, se puede visualizar el resultado, para ello se presenta el mapa con vista en planta dividido en los diferentes nodos que el programa ha sido capaz de detectar. Las figuras 4.18–4.20 muestran los resultados que se presentan con los valores de nodos obtenidos y silueta. Estos resultados son los obtenidos con los valores de parámetros $y = 1{,}5$ y $\Omega = 1{,}85$, valores que habían dado buenos resultados en los experimentos con el primer algoritmo creado y con el descriptor HOG.

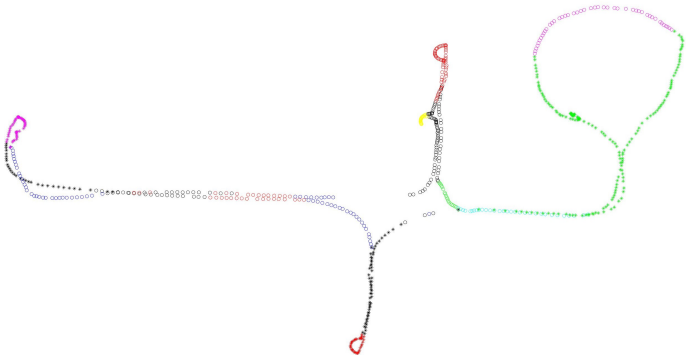
Visualmente se puede observar como los nodos están mejor definidos, no se dejan espacios con imágenes que no se consideran en ninguno de los nodos creados y el valor de silueta mejora. En general, los resultados muestran un aumento en la detección de nuevos nodos y una bajada de imágenes sin asignar. Los valores de silueta son similares a los obtenidos anteriormente o mejoran ligeramente. Para rutas creadas con

(a) 8 nodos y 0.2544 de silueta.

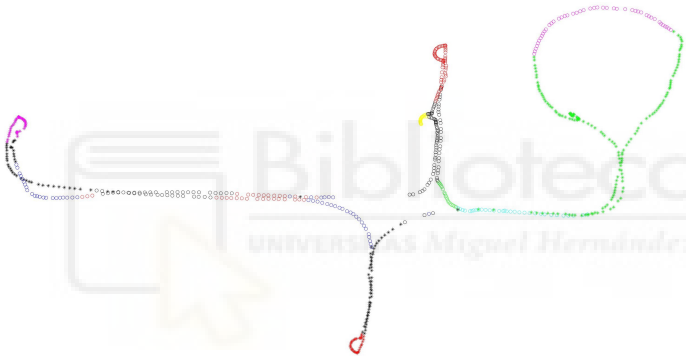
(b) 11 nodos y 0.3099 de silueta.

Figura 4.18: Mapas obtenido utilizando HOG, $y = 1{,}5$ y $\Omega = 1{,}85$ en la ruta de INNOVA. La primera subfigura muestra el mapa obtenido con el proceso sin mejoras, la segunda subfigura el mapa y los valores de nodos y silueta obtenidos con las mejoras del proceso.

menos imágenes como la de Saarbrücken, las mejoras no suponen mucho cambio en los resultados; pero en el caso de la ruta de Freiburg, que es creada con una mayor cantidad de imágenes, el resultado tanto visual como de silueta mejora considerablemente. Es preciso destacar la mejora en los resultados de Freiburg ya que con los cambios en el algoritmo se mejora el valor de silueta en todas las combinaciones de parámetros.

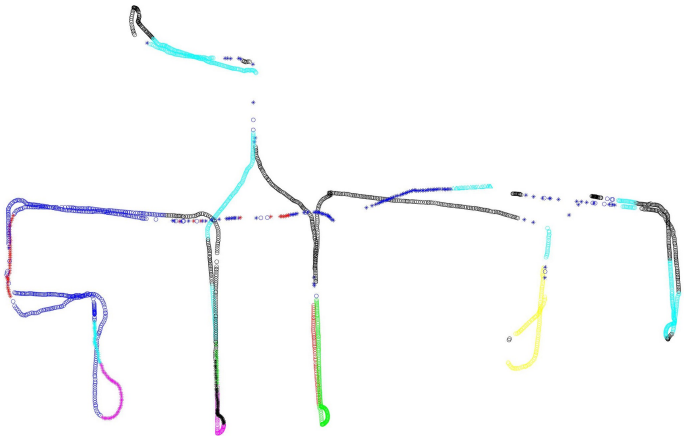


(a) 11 nodos y 0.2273 de silueta.

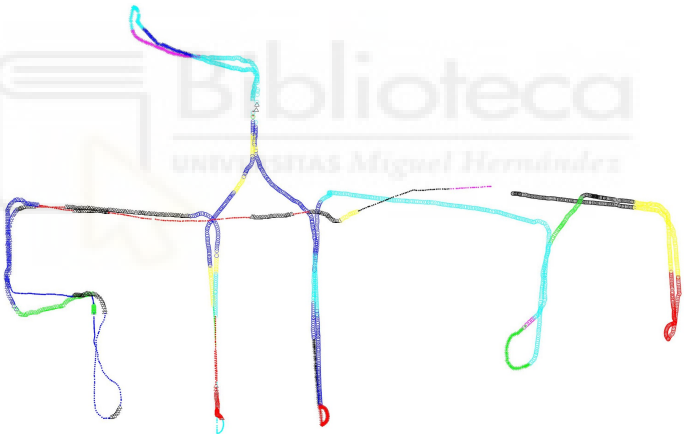


(b) 11 nodos y 0.2275 de silueta.

Figura 4.19: Mapas obtenido utilizando HOG, $y = 1{,}5$ y $\Omega = 1{,}85$ en la ruta de Saarbrücken. La primera subfigura muestra el mapa obtenido con el proceso sin mejoras, la segunda subfigura el mapa y los valores de nodos y silueta obtenidos con las mejoras del proceso.



(a) 14 nodos y -0.1628 de silueta.



(b) 24 nodos y 0.0835 de silueta.

Figura 4.20: Mapas obtenido utilizando HOG, $y = 1{,}5$ y $\Omega = 1{,}85$ en la ruta de Freiburg. La primera subfigura muestra el mapa obtenido con el proceso sin mejoras, la segunda subfigura el mapa y los valores de nodos y silueta obtenidos con las mejoras del proceso.

4.7. Conclusiones

En este capítulo se ha presentado un método y sus mejoras para crear mapas en entornos de interior de manera incremental, actualizando el modelo cada vez que una nueva imagen es tomada. El marco está basado en el desarrollo de un algoritmo de clustering incremental, presentado a lo largo del capítulo. Los experimentos se han realizado en entornos interiores por los que el robot ha navegado bajo condiciones reales,

incluyendo variación de iluminación y cambios introducidos por la actividad humana. Los robots con los que se han realizado los experimentos están equipados con un sistema de visión omnidireccional y la única información usada para construir el mapa jerárquico son las imágenes capturadas por el sistema. Para describir las imágenes se hace uso de dos métodos de descripción global de las imágenes: HOG y *gist*. La sección de experimentos muestra el rendimiento del algoritmo y la influencia de los principales parámetros en el resultado final. Además, se muestra una comparativa para evaluar los resultados obtenidos por el algoritmo propuesto con un algoritmo de clustering espectral que funciona *off-line*.

La precisión de estos métodos se evalúa calculado la silueta media. Para calcular la silueta se considera los descriptores de las imágenes como entidades y se estudia cuánto se parece a las entidades de su nodo y cuanto de diferente es al resto de los nodos. Primero, HOG demuestra tener mejores resultados; ya que la silueta media obtenida es similar a la silueta obtenida usando un método de clustering espectral absoluto, pese que a que el algoritmo propuesto va obteniendo información conforme se visita el entorno. De hecho, si los parámetros se ajustan adecuadamente, puede dar lugar a resultados de silueta muy parejos al clustering espectral. Los resultados muestran que en el caso de HOG es especialmente importante ajustar el parámetro y de forma adecuada, ya que tiene una fuerte influencia en el número final de clusters obtenidos. Además, los resultados obtenidos también muestran un deterioro en el rendimiento del algoritmo cuando la base de datos contiene un número excesivo de imágenes o cuando el entorno presenta características complejas (por ejemplo, muchas ventanas o muros de cristal o características que tiendan al *aliasing visual*. Por último, se presentan unos cambios y mejoras en el algoritmo que consideran estas características y mejoran el rendimiento del método presentado en entornos de gran extensión.

Los resultados deben ser tomados como satisfactorios ya que el algoritmo incremental propuesto empieza con un número reducido de imágenes y va actualizando el mapa (y los clusters) cada vez que una nueva imagen es capturada. Los algoritmos de clustering anteriores necesitan toda la información antes de comenzar con el proceso y poder calcular el conjunto de similitudes y diferencias entre descriptores. Adicionalmente, se puede concluir que el descriptor *gist* tiende a realizar estos mapas de forma menos robusta que con HOG, tanto en valor de silueta como en apreciación visual de separación entre los clusters calculados. Las siluetas obtenidas en los resultados que se obtienen con el clustering espectral son un poco mayores a las obtenidas con el proceso propuesto. A pesar de todo, el método de creación de mapas propuesto utilizando imágenes omnidireccionales y el descriptor HOG constituye la mejor opción para realizar mapas de manera incremental.

Dentro del mundo de la robótica móvil, estos trabajos abren la puerta a nuevos trabajos de investigación en la creación de mapas jerárquicos de manera incremental usando descriptores de apariencia global. Una vez se ha probado la eficacia y la solidez de los métodos en entornos reales de interior que incluyen actividad humana, variación de la iluminación artificial y la presencia de cambios en la posición de algunos objetos, el siguiente paso es mejorar el algoritmo y adaptar su uso para largos entornos de exterior. En este sentido, cabe destacar los posibles trabajos en la dirección de la adaptación

a cambios abruptos de las condiciones de iluminación y los cambios de estación, ya que parecen ser condiciones que pueden suponer peor impacto en los algoritmos de creación de mapas de forma visual.

Cuadro 4.4: Parámetros con impacto en el tamaño de los descriptores.

HOG	
Descriptor de posición	b_{hp}⇒ bins por histograma en el descriptor de posición.
	k_{hp}⇒ número de celdas horizontales en el descriptor de posición.
Descriptor de orientación	b_{ho}⇒ bins por histograma en el descriptor de orientación.
	k_{ho}⇒ número de celdas verticales en el descriptor de orientación.
	$dist_{ho}$⇒ distancia entre celdas verticales consecutivas.
Gist	
Descriptor de posición	m_{gp}⇒ número de filtros Gabor en el descriptor de posición.
	k_{gp}⇒ número de celdas horizontales en el descriptor de posición.
	r_{gp}⇒ modelos de resolución en el descriptor de posición.
Descriptor de orientación	m_{go}⇒ número de filtros Gabor en el descriptor de orientación.
	k_{go}⇒ número de celdas verticales en el descriptor de orientación.
	r_{go}⇒ modelos de resolución en el descriptor de posición.
	$dist_{go}$⇒ distancia entre celdas verticales consecutivas.

Cuadro 4.5: Símbolos usados durante el proceso de mapping incremental jerárquico.

Símbolo	Definición
I_q	Nueva imagen adquirida.
$\vec{d}_q$	Descripción de la nueva imagen.
N^C	Conjunto de nodos contenidos en el mapa.
C	Número actual de nodos.
Node Level Loop Closure	
N^*	Grupo de nodos que cumplen la condición Node Level Loop Closure.
N_l	Nodo evaluado en la condición Node Level Loop Closure.
$\vec{\mu}^{N_l}$	Descriptor medio de N_l (ec. 4.1).
$\sum_{d}^{N_l}$	Matriz de covarianza de N_l (ec. 4.1).
$\Delta_{\vec{d}_q}^{N_l}$	Distancia de Mahalanobis entre $\vec{d}_q$ y N_l (ec. 4.1).
$\mu_{ns}^{N_l}$	Distancia de Mahalanobis media (ec. 4.2).
$\sigma_{ns}^{N_l}$	Desviación estandard de las distancias de Mahalanobis (ec. 4.2).
x	Modifica ec. 4.2. Su valor depende del número de clusters.
Ω	*Parámetro ajustable*. Limita el valor mínimo de x.
Image Level Loop Closure	
I_i	Imagen detectada como más similar a I_q.
N_i	Nodo al que pertenece I_i.
I^{N^*}	Todas las imágenes contenidas en los nodos N^*.
$\vec{d}_k$	Descriptores de las imágenes I^{N^*}.
$dist_{eucl}^{\vec{d}_k}{}_{\vec{d}_q}$	Distancia Euclídea entre $\vec{d}_q$ y $\vec{d}_k$ (ec. 4.3).
$sim_{\vec{d}_q}^{\vec{d}_k}$	Similitud entre los descriptores $\vec{d}_q$ y $\vec{d}_k$ (ec. 4.4).
Condición de prominencia	
P_{I_i}	Valor de prominencia de la imagen candidata I_i (ec. 4.6).
$\mu(P_{I_{k^*}})$	Prominencia media de las imágenes candidatas (ec. 4.6).
γ	*Parámetro ajustable*. Limita la diferencia mínima en la ec. 4.6.
Fusión de nodos	
N_o	Nodo existente evaluado.
N_q	Nuevo nodo creado.
$\Delta_{N_q}^{N_o}$	Distancia Mahalanobis entre nodos N_q y N_o (ec. 4.8).
μ^{N_q}	Descriptor medio de N_q (ec. 4.8).
μ^{N_o}	Descriptor medio de N_o(ec. 4.8).
$\sum_{d}^{N_o}$	Matriz de covarianza de N_o (ec. 4.8).
y	*Parámetro ajustable*. Restringe la fusión de nodos (eq.4.9)

Cuadro 4.6: Parámetros que necesitan ser ajustados.

Parámetro	Definición	Valores
Ω	Limita el valor de x.	1.7, 1.85, 2, 2.15, 2.3
γ	Limita la diferencia en ec.(4.6).	Por experimentos $\gamma = 5$
y	Limita la fusión de nodos ec.(4.9).	0.75, 1, 1.25, 1.5, 1.75, 2, 2.25

4.9. Publicaciones relacionadas con este capítulo

Los principales resultados presentados en este capítulo están relacionados en la siguientes publicaciones:

- V. Román, L. Payá, S. Cebollada, O. Reinoso. Creating Incremental Models of Indoor Environments through Omnidirectional Imaging. Applied Sciences. Ed. MDPI. Vol 10(18),6480 (Septiembre 2020) [142]. **JCR-SCI Impact Factor: 2.679**, Quartile **Q2**.

    - Este artículo presenta la aplicación de mapping incremental presentada a lo largo de este capítulo. En él se han utilizado los descriptores HOG y *gist* para realizar esta tarea y se han optimizado los parámetros que influyen en el proceso. Se han utilizado imágenes omnidireccionales de las bases de datos COLD y de la universidad Miguel Hernández y se ha presentado los resultados de los experimentos, así como una comparación con otras técnicas ya conocidas.

5 Uso de redes siamesas para tareas de localización jerárquica y absoluta

5.1. Introducción

Como se ha expuesto en los capítulos previos la robótica móvil bebe del desarrollo de técnicas de visión por computador para mejorar su autonomía en el área de la navegación [139]. Tal y como se ha visto anteriormente, tradicionalmente la información recibida se ha descrito utilizando las técnicas de descripción local [52, 99, 153, 28, 91] o de descripción global [19, 79, 76, 95]. Sin embargo, estudios recientes han utilizando técnicas de aprendizaje profundo para crear nuevos descriptores. Por ejemplo, Xu et al. [181] y Leyva et al. [86] proponen unos descriptores de apariencia global basado en CNNs (Convolutional Neural Network) con los que obtener la pose del robot más probable. Da Silva et al. [45] proponen un trabajo en el que se utilizan descriptores basados en CNNs y cámaras omnidireccionales para conseguir soluciones a la localización y navegación de robots móviles.

Cada vez con mayor regularidad se están utilizando las técnicas de Inteligencia Artificial (IA) para aplicaciones en el campo de la robótica. De este modo, podemos encontrar trabajos en los ámbitos de la *Conducción autónoma* [154, 134, 124], *detección y reconocimiento de caras* [178, 71, 67], *reconocimiento y categorización de objetos* [120, 186, 49] y *tareas de mapping y localización* [170, 65, 148].

Entre las técnicas de Inteligencia Artificial más potentes y populares se encuentran las CNNs (Convolutional Neural Networks). A menudo se utilizan para tareas de localización y creación de mapas debido a su alto rendimiento. Estas redes están diseñadas para tener imágenes como entrada y sus estructuras entrenadas para extraer características locales, patrones o una descripción global de la imagen [40].

Ciertas arquitecturas de CNN son bastante reconocidas para tareas de reconocimiento de escenas. Por ejemplo, AlexNet [81], VGG16 [158], GoogleNet [167] o Netvland [9]. Durante los últimos años, las CNNs y la IA se han utilizado para diferentes tareas en el marco de la robótica móvil. Por ejemplo, en *mapping* [159, 113, 25], *localización* [179, 89, 30], *navegación* [188, 102] y/o *localización y creación de mapas simultáneos (SLAM)* [98, 94].

Para más información relacionada con los trabajos de aprendizaje profundo en tareas de robótica es posible consultar el estado del arte de Cebollada et al. [31].

La finalidad de este capítulo es completar las tareas de localización jerárquica y absoluta utilizando Redes Neuronales Siamesas (Siamese Neuronal Networks). Las redes siamesas son una arquitectura basada en redes convolucionales en las que dos imágenes son evaluadas al mismo tiempo y cuyo uso ha sido poco explorado en tareas de robótica móvil. El funcionamiento se describe en la sección 5.3.

Durante este capitulo se expone una comparativa de diferentes redes siamesas utilizadas para obtener la estructura que mejor resultado proporcione. La tarea es una localización en entornos heterogéneos en los que el principal problema viene dado por cambios en la iluminación de la escena.

El problema derivado de los cambios de iluminación es un gran problema en las tareas de navegación de robots móviles. Sakkos et al. [151] intentan afrontar los cambios en la intensidad lumínica del entorno utilizando Inteligencia Artificial. Pero este no es el único problema derivado de trabajar en entornos dinámicos. Otros problemas pueden ser causados por la aparición de personas en la escena, cambios en el mobiliario, etc. Se espera que la Inteligencia Artificial y en concreto las redes neuronales ayuden a mitigar los posibles problemas derivados. La esencia de este capítulo es estudiar las redes siamesas y evaluar qué estructura y parámetros dan lugar a la red más robusta para resolver de forma eficiente la tarea de localización bajo las condiciones presentadas.

Como en el resto de los trabajos de la tesis, la cámara omnidireccional es la única fuente de información usada. Las imágenes utilizadas en los experimentos se encuentran en la base de datos de interior [135]. Las imágenes omnidireccionales se transforman en imágenes panorámicas y la red siamesa recibe a su entrada, de manera simultánea, dos imágenes.

El capítulo se estructura con una introducción a la localización visual en el apartado 5.2. A continuación, en la sección 5.3, se exponen las arquitecturas empleadas y los detalles sobre la realización del entrenamiento de las redes. En 5.4 se muestran los experimentos y resultados obtenidos. Después se presentan diferentes conclusiones en 5.5.

5.2. Localización Visual

5.2.1. Localización jerárquica

Esta sección se focaliza en explicar la localización jerárquica del robot móvil utilizando información visual. Para estos trabajos se asume que el robot dispone de

un modelo visual del entorno. Para obtener este modelo, previamente el robot se ha movido por el área capturando imágenes omnidireccionales a lo largo de una trayectoria. Primero, las imágenes se transforman a formato panorámico. Como resultado se tiene un conjunto de imágenes $\{f_1, f_2, ..., f_j\}$ con tamaño 128x512. Estas imágenes han sido capturadas desde j puntos de los que se conoce su pose $\vec{P}_i = (x_i, y_i, \theta_i), i = 1, ..., j$. Adicionalmente, la habitación en la cual la imagen ha sido capturada también se almacena $\vec{R}_i = (r_i), i = 1, ..., j$, donde r_i es una etiqueta que identifica el número de habitación desde la que se capturó la imagen f_i. La trayectoria pasa por diferentes áreas con información visual diferenciada y corresponden a pasillo, diferentes oficinas, biblioteca, aseo...

Teniendo en cuenta esta información, el modelo está compuesto por el conjunto de imágenes y sus poses junto con la habitación en la que las imágenes han sido capturadas $\{(f_1, \vec{p_1}, r_1), (f_2, \vec{p_2}, r_2), ..., (f_j, \vec{p_j}, r_j)\}$.

Una vez el modelo ha sido construido, el problema de localización se puede llevar a cabo siguiendo diferentes estrategias. Se ha estudiado un método de localización jerárquica y una estrategia de localización absoluta. Por un lado, desde el punto de vista de la localización jerárquica el problema se resuelve determinando primero la habitación de captura y a continuación se trata de estimar la pose de entre las imágenes que en el modelo están en la habitación estimada. Por otro lado, el problema de localización absoluta consiste en estimar la pose del robot comparando con todas las imágenes del modelo. Ambos enfoques se resuelven sin información métrica y sin información previa de la posición o habitación anterior. Toda la tarea se realiza con información visual pura.

Para resolver el problema, la información visual se condensa utilizando los descriptores globales. Gracias a descriptores holísticos vistos a lo largo de esta tesis o a herramientas de inteligencia artificial es posible reducir la imagen a un vector denominado descriptor. La localización absoluta se realiza como un problema de image retrieval, comparando pares de imágenes o descriptores. La imagen test se compara con imágenes almacenadas en el modelo y el punto de captura de la imagen más similar se selecciona como la pose del robot. Siguiendo la idea de la localización jerárquica, el modelo se divide en diferentes capas y cada capa tiene un nivel de granularidad diferente. En el caso estudiado en esta aplicación, en una primera capa se determina a qué estancia pertenece la imagen de test, y a continuación se aproxima con un grado de precisión mayor utilizando la técnica de comparación de pares de imágenes o vectores, usando sólo las imágenes del modelo contenidas en la habitación seleccionada.

Para el estudio presentado se hace uso de las redes siamesas para resolver el problema de localización. Para ello, pares de imágenes y su etiqueta de similitud se introducen en la red para su entrenamiento. Primero, la red 'aprende a resolver el problema' en la fase de entrenamiento, después la red se puede usar para predecir la habitación o pose de la nueva imagen. Las siguientes subsecciones detallan los métodos propuestos para resolver el problema de localización.

5.2.2. Herramientas de Deep Learning

Esta sección se centra en la descripción de las herramientas de *deep learning* (aprendizaje profundo) para describir globalmente el conjunto de imágenes. El propósito de esta técnica es obtener un descriptor de características único que represente cada imagen. El descriptor debe ser capaz de describir la escena resaltando características que hagan reconocer que dos escenas son similares. Estas características deben ser lo suficientemente invariantes temporalmente para que se reconozcan similitudes cuando hay cambios en la escena. Con la idea de las redes siamesas, un par de imágenes son descritas simultáneamente y sus descriptores son evaluados conjuntamente. De esta evaluación se obtiene un valor de disimilitud entre las imágenes.

Teniendo en cuenta la fase de descripción, se utiliza como base una red CNN. Este método viene de otras técnicas de aprendizaje profundo para tareas de clasificación, como en los trabajos de Krizhevsky et al. [81]. Además, en los últimos años se han usado técnicas de aprendizaje profundo para obtener descriptores visuales mas complejos. Cebollada et al. [36] proponen descriptores holísticos basados en CNN para realizar tareas de localización en modelos topológicos y estudian su comportamiento ante variaciones en la iluminación, Xu et al. [181] y Leyva et al. [86] proponen este tipo de técnicas para obtener la posición más probable de un robot. Adicionalmente, Ballesta et al. [17] estudian una aplicación de tareas de localización utilizando CNNs y capas de regresión como descriptores de apariencia global.

Sin embargo, todos estos trabajos evaluan cada imagen de manera individual. El objetivo de nuestros estudios es utilizar un tipo especial de arquitectura CNN en el que dos redes trabajan en paralelo y comparten sus pesos para obtener el descriptor de dos imágenes simultaneamente (figura 5.1). Este tipo de redes se les conoce como redes siamesas. Un primer análisis que se puede usar como referencia son los trabajos de Leyva et al. [87, 88], en ellos, se realiza un primer estudio de esta técnica para reconocimiento de lugares visitados en entornos de jardín

Las redes siamesas permiten introducir dos entradas al mismo tiempo. Cada imagen de entrada es trabajada de manera individual, pero los modelos comparten sus pesos. Este modelo proporciona dos vectores de salida (uno por cada imagen de entrada) y estos descriptores se comparan. La red aprende a discernir cuando estos vectores son similares o disimilares en la fase de comparación, la fase de comparación utiliza etiquetas y la función de pérdida (*loss function*).

5.3. Arquitectura y entrenamiento de las herramientas de Deep Learning

La estructura de una red CNN simple (no siamesa) usada para tareas de clasificación puede ser separada en dos fases diferentes [36] : la fase de aprendizaje de características y la fase de clasificación. Las características se aprenden utilizando un conjunto de capas convolucionales mientras que la tarea de clasificación se realiza utilizando capas softmax y totalmente conectadas. En la primera parte de la red se extraen

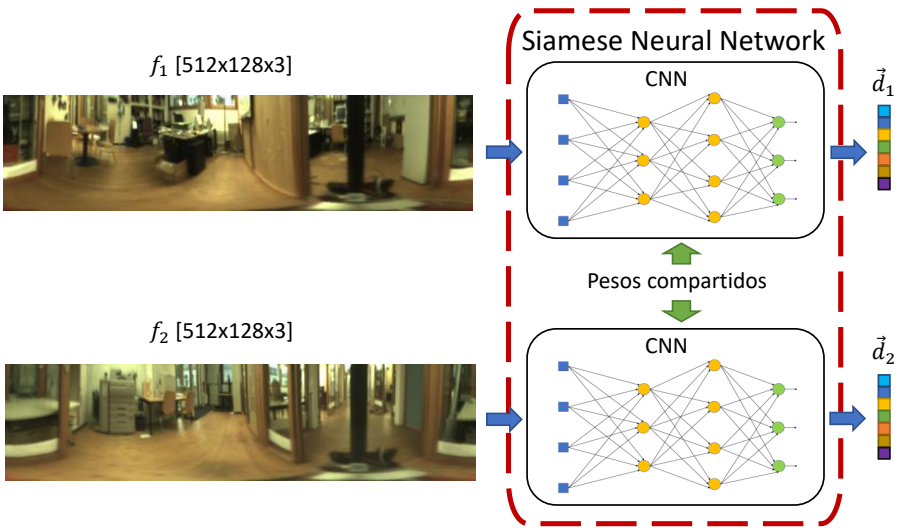


Figura 5.1: Esquema de red siamesa con dos imágenes de entrada y dos descriptores como salida.

las características principales y después se realiza la tarea de clasificación. Sin embargo, en este estudio no es interesante la tarea de clasificación; por esa razón, solo se utiliza la serie de capas convolucionales para la extracción de características, pero la parte de clasificación se realiza de otra manera. Tras las capas de extracción de características, el vector se reduce con una serie de capas totalmente conectadas. La fase de extracción de características ofrece una matriz que se reduce a un único vector en la fase de reducción. Esta fase será conocida como fase de escalado y estará formada por una capa softmax y unas pocas capas totalmente conectadas.

Teniendo en cuenta esta idea y el objetivo de este trabajo en el que las redes siamesas comparten pesos y aprenden de manera conjunta, la fase de extracción de características se compone por dos redes que comparten pesos. Después, los descriptores se reducen y comparan en la siguiente etapa, la fase de escalado, tal y como se muestra en la figura 5.2.

Las ramas siamesas dan como resultado dos vectores $(\vec{d_0}, \vec{d_1})$ (uno por imagen de entrada), estos descriptores se comparan utilizando la distancia Euclidea en la fase de comparación $(dist(\vec{d_0}, \vec{d_1}) = \|\vec{d_0} - \vec{d_1}\|^2)$. De este modo, durante el entrenamiento, los pesos de la red se van actualizando para obtener descriptores más ajustados. Los descriptores se comparan y la distancia entre ellos y la etiqueta de similitud $(y, 1 : disimilar, 0 : similar)$ se usa como información para la función de pérdida. En nuestro caso, la función de pérdida utilizada es *Contrastive Loss Function* (ec. 5.1).

$$L(d_0, d_1) = (1 - y) * dist(\vec{d_0}, \vec{d_1})^2 + y * max(\alpha - dist(\vec{d_0}, \vec{d_1}), 0)^2 \qquad (5.1)$$

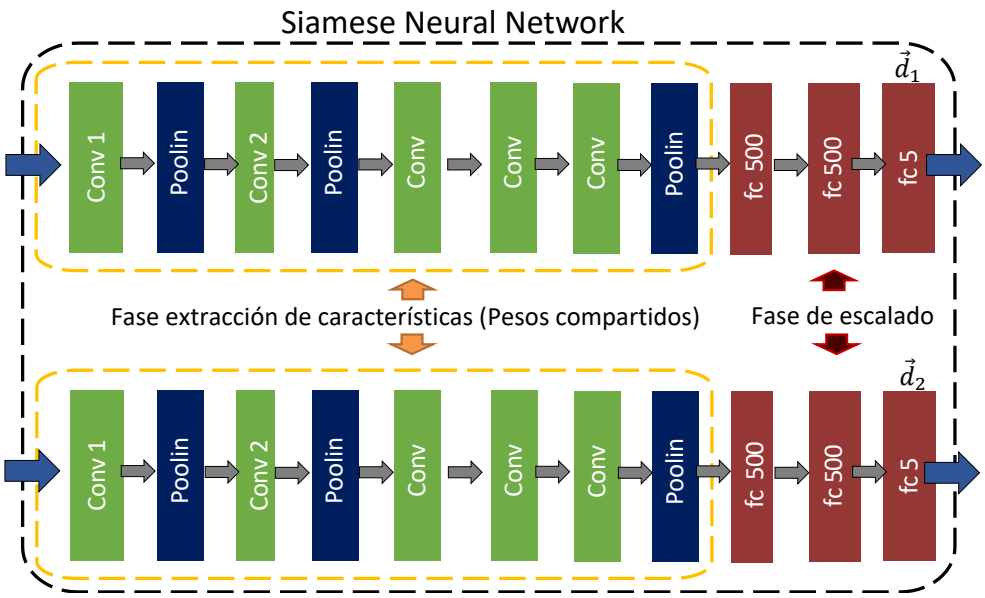


Figura 5.2: Esquema de las fases de extracción de características y de escalado en la red siamesa.

5.3.1. Parámetros y redes

Para la fase de aprendizaje de características se han evaluado diferentes redes. Como entrada a las capas de escalado se tiene el último descriptor obtenido en las capas convolucionales de la fase de aprendizaje de características. Como arquitectura base en la fase de aprendizaje de características se usan estructuras predeterminadas como Alexnet [81], DenseNet [61], VGG11, VGG13, VGG16 y VGG19 [158]. De manera adicional, se estudian dos redes simples creadas por tres capas conv2d. En la tabla 5.1 se presentan las capas de cada representación. Las capas de activación ReLU no se representan por falta de espacio, pero se utilizan tras cada capa conv2d. Las diferentes estructuras se evaluan y se presentan los resultados en la sección 5.4.

En la etapa de escalado, también se testarán diferentes estructuras. De forma general se van a utilizar tres capas totalmente conectadas (*fully connected layers*), pero se evalúan diferentes versiones con diferentes números de neuronas. Las diferentes versiones evaluadas se representan en la tabla 5.2.

Adicionalmente se puede ajustar otros parámetros que tienen influencia en el entrenamiento, estos parámetros son conocidos como hiperparámetros. Con el ánimo de obtener las mejores redes neuronales siamesas, se ha estudiado la influencia de varios de estos parámetros en nuestros entrenamientos. Los hiperparámetros que se han evaluados son los siguientes:

Tamaño de lote (batch size): El tamaño de lote representa el numero de muestras procesadas antes de actualizar el modelo. El tamaño de lote debe ser mayor

Cuadro 5.1: Configuraciones utilizadas en la fase de extracción de características.

Simple 1	**Simple 2**	**AlexNet**[81]	**DenseNet**[61]	**VGG11**[158]	**VGG13**[158]	**V**
input (128 X 512 imagen RGB)						
conv2d-3 conv2d-8 conv2d-16	conv2d-3 conv2d-16 conv2d-32	conv2d-64	conv2d-112	conv2d-64	conv2d-64 conv2d-64	
maxpool	maxpool	maxpool	maxpool	maxpo		
		conv2d-192	conv2d-56 x 6	conv2d-128	conv2d-128 conv2d-128	
		maxpool	averagepool	maxpo		
		conv2d-384 conv2d-256 conv2d-256	conv2d-28 x 12	conv2d-256 conv2d-256	conv2d-256 conv2d-256	
		maxpool	averagepool	maxpo		
			conv2d-14 x 24	conv2d-512 conv2d-512	conv2d-512 conv2d-512	
			averagepool	maxpo		
			conv2d-7 x 16	conv2d-512 conv2d-512	conv2d-512 conv2d-512	
			averagepool	maxpo		
FC-500						
FC-500						
FC-5						

* En color azul se presentan las capas que corresponden a las capas de escalado.

Las redes VGG tienen su versión **bn en las que tras cada capa conv2d se normaliza el vector con una capa BatchNorm2d.

Cuadro 5.2: Configuración de las redes en la fase de escalado.

versión 1	**versión 2**	**versión 3**	**versión 4**
FC-500	FC - 500	FC - 1000	FC - 4096
FC - 500	FC - 100	FC - 1000	FC - 4096
FC - 5	FC - 10	FC - 10	FC - 1000

o igual a uno y menor o igual que el numero de muestras en la base de datos de entrenamiento. Se experimenta con diferentes tamaños:[8, 16,...,256,512].

Épocas (Epoch): El número de épocas es el número de veces que se estudia el conjunto de datos de entrenamiento. Cuanto mayor es este número, mayor tiempo de entrenamiento de la red. Pero un número muy alto de epoch puede sobre-entrenar la red. Este parámetro toma diferentes valores durante los experimentos. La evaluación se realiza con valores de epoch desde los 5 a los 30 epoch.

Porcentaje de imágenes: La red necesita pares de imágenes para ser entrenada. Estos pares de imágenes pueden estar compuestos por imágenes tomadas en la misma habitación o tomadas en diferentes habitaciones. Ambas combinaciones son necesarias durante el entrenamiento para aprender similitudes y diferencias entre muestras. El porcentaje de imágenes etiquetadas como iguales/diferentes cambia en los experimentos. Se toma porcentajes de todo tipo: 50 %-50 %, 10 %-90 %, 20 %-80 %...

Tasa de aprendizaje: Es un valor escalar positivo que controla cuánto cambiar el modelo en respuesta al error estimado cada vez que se actualizan los pesos del modelo. Este hiperparámetro se mantiene constante en 0.01.

Momentum: Valor escalar de 0 a 1 que indica la contribución de cada paso en la actualización de parámetros de la iteración anterior a la iteración actual. Un valor de 0 significa que no hay contribución del paso anterior, mientras que un valor de 1 significa una contribución máxima del paso anterior. Se ha decidido mantener este valor constante en 0.9

Optimizador: La esencia de la red neuronal es minimizar la pérdida, una pérdida pequeña significa que el modelo funciona mejor. Para minimizar la pérdida es necesario utiliza un función de optimización. Los optimizadores son algoritmos o métodos usados para cambiar los atributos de la red neuronal como pesos y sesgos para reducir las pérdidas. Durante los experimentos se ha usado especialmente el optimizador Stochastic Gradient Descent (sgd).

5.3.2. Data Augmentation

Adicionalmente, se va a ensayar la utilización de nuevas técnicas de *data augmentation*. Se propone un método para tener más información para entrenar la red y mejorar el desempeño de la red. Al obtener artificialmente más imágenes de entrenamiento el número de posibles pares de imágenes en la *fase de entrenamiento* aumenta, y se pueden estudiar problemas que pueden surgir en la *fase de test* y aprender de este modo a ser robusto ante estos efectos. La técnica de *data augmentation* consiste en obtener más imágenes de entrenamiento modificando las que ya se tiene con diversos efectos. Nuestra propuesta de *data augmentation* se centra en replicar los efectos visuales adversos que pueden surgir en una situación de trabajo real, incluyendo cambios de iluminación. Para ello se editan regiones locales de la imagen simulando efectos de luces, reflejos y sombras producidos por focos de luz en diferentes ángulos. Además,

también se añaden cambios globales en la iluminación y se edita toda la imagen alterando el contraste, la nitidez, la iluminación global... Por último, también se tienen en cuenta otros efectos no relacionados con la iluminación pero que también tienen cabida en aplicaciones de localización en condiciones reales. Los trabajos de Cabrera et al. [26] y Sakkos et al. [152] demuestran que el uso de *data augmentation* en CNNs aumenta la efectividad de estas. En resumen, el proceso de *data augmentation* consiste en aplicar efectos visuales sobre la imagen original y crear artificialmente nuevas imágenes para entrenar la red.

A continuación, se presentan los efectos que se han aplicado a las imágenes para obtener el conjunto de entrenamiento aumentado.

Efectos Locales: Se trata de reproducir focos de oscuridad o luz en áreas específicas de la escena. Llamamos a esto cambios locales de iluminación ya que hacemos influir en pequeños parches o zonas de la imagen. La forma de los focos varía significativamente para que se adapte a todas las posibles formas.

Se utilizan formas circulares para simular luces de bombillas o formas cuadradas y trapezoidales para simular reflejos o ventanas. También se edita la intensidad de la región para simular el foco de luz, cuyo valor de intensidad varía aumentando o disminuyendo su valor respecto de la zona para reproducir de la mejor manera un efecto real de foco de luz o sombra. Para replicar de forma realista el efecto de desvanecimiento de influencia, la intensidad de la luz/oscuridad desciende gradualmente desde el centro del foco generado hacia los extremos, simulando la atenuación de un caso real.

El tamaño de las formas y la posición de estas se selecciona de manera aleatoria y de esta manera se simulan diferentes formas y grados de variación de intensidad. En nuestros experimentos, los efectos locales se construyen con un radio entre 15 y 40 píxeles y la intensidad añadida se degrada desde entre $+/-$ 160 y $+/-$100 hasta 5. Los distintos efectos y formas se pueden visualizar en la figura 5.3.

Iluminación global: Variaciones de la iluminación en la imagen de forma global también pueden ocurrir en algunos casos. Para modelar estos cambios de iluminación es necesario alterar todos los píxeles de la imagen, en vez de en una pequeña región de la imagen como se hace con los efectos locales. Se añade a todos los píxeles un valor constante y se modela un efecto de mayor iluminación, o se substrae de todos los píxeles un valor constante para crear un efecto de mayor oscuridad. El efecto se aplica siguiendo la siguiente ecuación:

$$I_s = I \pm c$$

Donde I_s es la nueva imagen, I es la imagen original y c es el valor constante que se añade o se resta de los píxeles de la imagen para crear el efecto de luz u oscuridad. Se generan diferentes valores de intensidad, el parámetro c varía de 35 a 75, dependiendo de la imagen. Las figuras 5.4(b) y 5.4(c) muestran este efecto.

(a) Efecto de iluminación con bombilla.

(b) Efecto de oscuridad con bombilla.

(c) Efecto de iluminación con cuadrado.

(d) Efecto de iluminación con trapezoide.

Figura 5.3: Efectos locales individuales para un 'data augmentation' basado en iluminación.

Cuadro 5.3: Máscaras para el aumento del efecto 'sharpness' y 'blurring'.

Efecto 'sharpness'	**Efecto 'blurring'**
$m_{sh} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$m_{bl} = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

Aumento de Sharpness/Blurring: Traducidos del inglés, el efecto 'sharpness' hace referencia a la nitidez de la imagen mientras que el efecto 'blurring' provoca cierto desenfoque de la escena. Busca tener unos bordes más nítidos y resaltados y contribuirá a tener una mejor separación entre los objetos y el fondo de la imagen. Por el contrario, el efecto blurring provoca un desenfoque que simula el posible efecto causado por una baja iluminación y/o un movimiento de la cámara mientras realiza la captura. Ambos efectos se incorporan al 'data augmentation' propuesto. Pueden ser seguidos en las figuras 5.4(d) y 5.4(e). Ambos efectos se pueden conseguir convolucionando la imagen con las máscaras de la tabla 5.3.

Variación en el contraste: El contraste de una imagen juega un rol importante en resaltar diferentes objetos en la escena. Imágenes con un bajo contraste suelen verse más suaves y con menos sombras y reflejos. Se propone este efecto dentro de nuestro 'data augmentation' para mejorar la robustez del entrenamiento. El contraste se modifica siguiendo la siguiente expresión:

$$I_s = 64 + c * (I - 64)$$

Donde I_s es la imagen resultante, I la imagen original y c es el factor de contraste. Para $c > 1$ el contraste de la imagen crece mientras que con $c < 1$ el contraste de la imagen decrece.

Además, otro efecto que se añade al 'data augmentation' propuesto es la ecualización. Este efecto, distribuye uniformemente los valores del histograma, lo que permite otra manera de aumentar el contraste. Se utilizan ambos métodos para variar el contraste de la imagen, tanto a través de la expresión anterior como con la ecualización. La figura 5.4(f) muestra este efecto obteniendo una variación del contraste empleando la fórmula y $c > 1$, con lo que se ha aumentado el contraste.

Cambios en la saturación: La saturación de la imagen se relaciona con la intensidad del color. Cuanto menor es la saturación, menos colorida es la imagen, llegando a parecer una imagen en escala de grises si la saturación es muy baja. Por el contrario, es posible obtener colores más vivos cuando la saturación de los colores crece. Este efecto puede simular situaciones en las que hay cambios de iluminación muy significativos.
La saturación puede editarse cambiando la imagen de RGB a HSV, después de esto, se puede editar directamente el canal de saturación. Suponiendo que la saturación se escala con el parámetro c, si el atributo de saturación se escala con $c > 1$ consigue una imagen con los colores más saturados, si $c < 1$ la saturación baja. El efecto puede verse en la figura 5.4(g).

Rotación: La imagen original es omnidireccional, esto significa que cubre 360 grados alrededor del plano del suelo. Por esta razón, la imagen puede rotarse sin perder información. Este efecto simula la posibilidad de tener el robot en una misma posición pero con una orientación diferente. Este efecto no tiene relación con efectos de iluminación como todos los expuestos hasta ahora, pero puede aumentar notoriamente el rendimiento, ya que se obtendrán situaciones que con mucha probabilidad se puede dar en la futura tarea de localización. La figura 5.4(h) muestra una rotación de 115 grados. Durante el 'data augmentation' se han realizado rotaciones aleatorias de entre 10 y 350 grados.

Cambios combinados: Por último, se han combinado efectos para obtener un 'data augmentation' más profundo. Pero no todos los efectos se han combinado conjuntamente. Los efectos de iluminación global y los efectos locales se han combinado en todas las variantes (por ejemplo, oscuridad global combinada con un trapezoide iluminado, efecto de oscuridad global combinado con un efecto local de bombilla oscura...). Además, también se han combinado efectos locales. El efecto de bombilla o círculo se combina con un cuadrado, el efecto de trapecio se combina con otro circulo... Y estas combinaciones pueden ser de luz+luz, luz+oscuridad y oscuridad+oscuridad; tres efectos de bombilla pueden aparecer de manera combinada en la misma imagen. Finalmente, el efecto de rotación se combina individualmente con todos los efectos descritos anteriormente.

5.3.3. Entrenamiento de la red y resolución del problema de localización

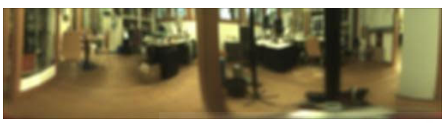
Para el entrenamiento, en un primer momento el robot crea un modelo del entorno. El modelo almacena la imagen, la posición donde ha sido tomada y la habitación

(a) Imagen original.

(b) Iluminación global.

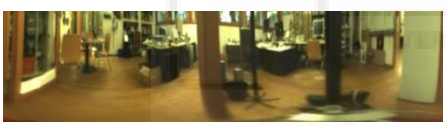
(c) Oscuridad global.

(d) Efecto sharpness.



(e) Ejecto blurring.

(f) Efecto de variación del contraste.



(g) Efecto de cambios en la saturación.

(h) Efecto rotación.

Figura 5.4: Efectos globales aplicados para el 'data augmentation'.

$\{(f_1, \vec{p_1}, r_1), (f_2, \vec{p_2}, r_2), ..., (f_j, \vec{p_j}, r_j)\}$.

Una vez el modelo ha sido construido, el problema de localización puede realizarse siguiendo diferentes estrategias. Durante nuestros trabajos se han propuesto dos alternativas para abordar el problema de localización. Por tanto, se presentan la localización jerárquica y la localización absoluta.

Por un lado, la **localización jerárquica** está basada en dos pasos. Primero, se recupera la habitación donde la imagen se ha capturado y a continuación, se refina la posición del robot dentro de la habitación elegida. Por otro lado, la **localización absoluta** se realiza comparando la nueva imagen con todas las imágenes del modelo y eligiendo la más similar.

Para solucionar el problema se entrena una red que resuelva el problema. Para el entrenamiento de la red, se introducen pares de imágenes y su etiqueta de similitud.

Cuadro 5.4: Ejemplo de pares con su valor de etiqueta en la tarea Room Retrieval.

Par	Valor de etiqueta
$I_1 - I_2$	0
$I_1 - I_3$	0
$I_1 - I_4$	1
$I_1 - I_5$	1
$I_4 - I_5$	1
$I_5 - I_6$	0

Primero la red siamesa 'aprende a resolver el problema' en la fase de entrenamiento, después las redes pueden ser utilizadas para predecir la similitud entre pares de imágenes. En las subsecciones 5.3.4 y 5.3.5, se detallan los diferentes pasos programados para resolver el problema de localización.

Pero entrenar una red neuronal desde cero es difícil, necesita mucha experiencia y conocimiento de la arquitectura de las redes, una gran cantidad de información y un enorme coste computacional. Debido a estas restricciones, se adapta una CNN preentrenada a cada una de las ramas de la red siamesa. De este modo, se obtiene una red neuronal siamesa que se reentrenará para realizar la tarea deseada, pero los pesos iniciales no son aleatorios sino que hay un preentrenamiento previo; esta técnica es conocida como 'transfer learning'. Como se ha presentado en la sección 5.3.1, se pueden utilizar diferentes CNNs como base para las redes siamesas. Las redes elegidas son seleccionadas con un preentrenamiento, de este modo al principio del entrenamiento los pesos de la red se mantienen de un entrenamiento anterior y no son ajustados desde cero. Posteriormente, en el reentrenamiento, los pesos que han sido aprendidos de la CNN original se adaptan a la nueva tarea para la que está siendo entrenada. La técnica de 'transfer learning' es conocida y se ha utilizado con anterioridad en trabajos de robótica móvil [26, 32].

5.3.4. Resolución del problema Room Retrieval

El objetivo principal de esta tarea es recuperar la habitación en la que se captura la nueva imagen. La *fase de entrenamiento* se realiza con las imágenes de la base de datos. Estas imágenes se introducen en la red por parejas y se etiquetan con 0 si pertenecen en la misma habitación y con 1 si no pertenecen a la misma habitación. Esta elección de etiquetas se puede observar en la figura 5.5 y la tabla 5.4. El ratio de pares de imágenes misma habitación / distinta habitación en la *fase de entrenamiento* puede variar.

En la *fase de test* se evalúan pares de imágenes y tras realizar los cálculos se obtiene una etiqueta con el grado de disimilitud entre las imágenes. La red etiqueta los pares de imágenes con un número entre 0 y 1, si el resultado está por debajo de 0.5 se interpreta que las imágenes han sido capturadas en la misma estancia.

Las imágenes del experimento se obtienen de la base de datos [135]. Esta base de datos, utilizada también durante los experimentos del capítulo 4, recoge diferentes

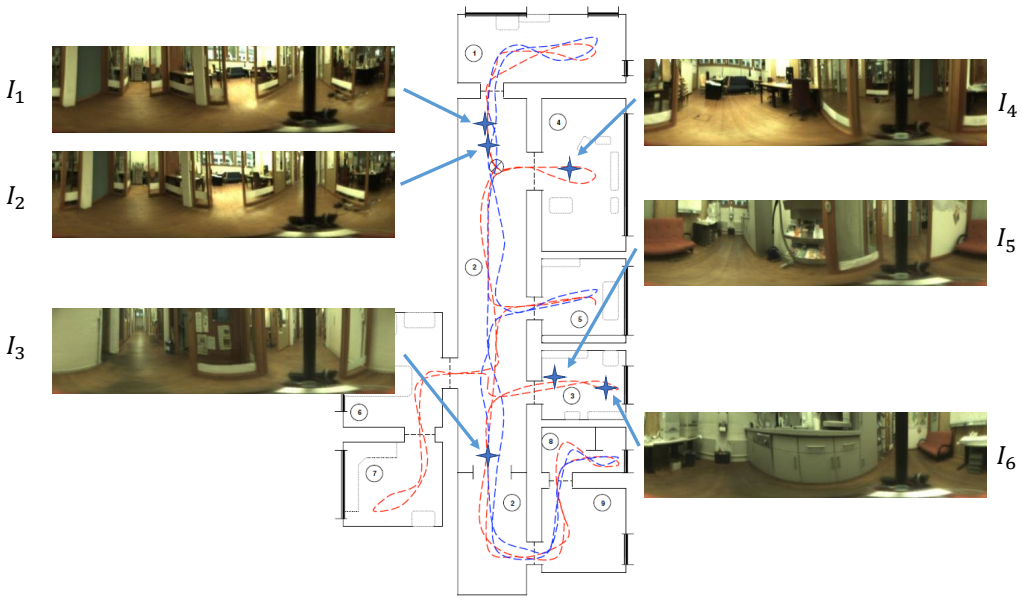


Figura 5.5: Ejemplos de imágenes para la explicación del etiquetado.

Cuadro 5.5: Habitaciones en la trayectoria Freiburg de la base de datos COLD [135].

Nombre en la base de datos	Habitación	Máxima distancia entre imágenes
1P0-A	Oficina individual	3.0181 m
2P01-A	Oficina compartida 1	3.6886 m
2P02-A	Oficina compartida 2	2.3537 m
CR-A	Pasillo	18.9940 m
KT-A	Cocina	4.3956 m
LO-A	Oficina larga	3.1061 m
PA-A	Área de impresoras	3.5094 m
ST-A	Zona de escaleras	2.2909 m
TL-A	Baño	2.9762 m

trayectorias de robots móviles. Las trayectorias se obtienen en días diversos y pasan por diferentes habitaciones. La tabla 5.5 recoge las diferentes habitaciones de la trayectoria con la que se ha trabajado, así como la distancia máxima entre imágenes de la misma habitación.

Para la *fase de entrenamiento* se han usado 8486 imágenes. Las imágenes están distribuidas uniformemente entre las habitaciones y se utilizan imágenes de nublado,

Cuadro 5.6: Ejemplo de pares con su valor de etiqueta en la tarea localización absoluta.

Par	Distancia euclídea	Valor de etiqueta
$I_1 - I_2$	0.33	$\frac{0,33}{18,99} = 0,017$
$I_1 - I_3$	12.82	$\frac{12,82}{18,99} = 0,675$
$I_1 - I_4$	-	1
$I_1 - I_5$	-	1
$I_4 - I_5$	-	1
$I_5 - I_6$	2.48	$\frac{2,48}{18,99} = 0,131$

Donde:

18.99 es la máxima distancia entre dos imágenes que están en la misma estancia

soleado y de noche durante el entrenamiento. Las trayectorias escogidas en la *fase de entrenamiento* son: *COLD-Freiburg Part A Path 2 Cloudy 3, Freiburg Part A Path 2 Night 1, Freiburg Part A Path 2 Sunny 3*. Por su parte, para la *fase de test* se han evaluado 7000 pares de imágenes tomadas en el mismo edificio pero en momentos diferentes. Al ser tomadas en otro momento las imágenes son del mismo entorno pero no captadas en los mismos puntos exactos ni con las mismas condiciones visuales. En esta fase se estudia también la localización de imágenes con variaciones de iluminación. Las imágenes evaluadas en la *fase de test* se toman de las trayectorias: *COLD-Freiburg Part A Path 2 Cloudy 2, Freiburg Part A Path 2 Night 2, Freiburg Part A Path 2 Sunny 2*. Al realizar el test de localización con imágenes tomadas en puntos distintos, con cambios visuales en el entorno y con cambios en la iluminación se asemeja bastante a las condiciones desafiantes que un robot móvil puede sufrir para localizarse.

5.3.5. Resolución del problema de localización absoluta

La resolución del problema de localización de manera absoluta implica tener información de toda la trayectoria y en un solo paso estimar la pose del robot. La *fase de entrenamiento* se realiza con pares de imágenes de la base de datos etiquetadas con números entre 0 y 1. Al contrario que en la resolución anterior, en este caso las etiquetas no son valores binarios y toman valores reales. El valor de la etiqueta se obtiene de normalizar la distancia euclídea entre los pares de imágenes. Los pares que tienen imágenes tomadas en diferentes estancias tienen 1 como valor de etiqueta y los pares con imágenes de la misma estancia tienen como etiqueta un valor normalizado de la distancia euclídea entre las posiciones donde se tomaron las imágenes. El valor se obtiene dividiendo la distancia euclídea entre las posiciones entre el valor máximo de distancia entre imágenes que pueden estar en una misma estancia, de este modo las imágenes más separadas toman el valor 1 y 0 si las imágenes están tomadas en el mismo punto. La forma de etiquetado de esta resolución se explica con la figura 5.5 y la tabla 5.6.

Después de que la red haya sido entrenada se puede realizar la *fase de test*. El robot toma una imagen en el instante t desde una pose desconocida. La nueva

imagen i_t se evalúa con todas las imágenes del modelo y cada comparación calcula un porcentaje de disimilitud entre imágenes. Una vez la nueva imagen se compara con todas las imágenes de la trayectoria modelo se elige aquella con el valor más bajo y se considera como la pose en la que el robot ha capturado la nueva imagen.

La *fase de entrenamiento* se realiza siguiendo dos perspectivas diferentes. Primero, siguiendo la idea del problema Room Retrieval, el entrenamiento se prepara usando imágenes de las tres variantes de iluminación (*cloudy*, *night* y *sunnny*), con lo que las mismas 8486 imágenes que se usan para entrenar el problema Room Retrieval se utilizan para entrenar un modelo que realice una localización absoluta. En segundo lugar, se estudia el caso en el que únicamente se ha tomado de manera previa imágenes un día nublado (*cloudy*). En este caso únicamente se dispone de 2778 imágenes COLD-Freiburg Part A Path 2 Cloudy 3 y se usa el proceso de *data augmentation* explicado en la subsección 5.3.2, lo que mejorará el entrenamiento. La técnica de *data augmentation* aumenta el número de imágenes de entrenamiento a 44448. Para estudiar cuánto es posible mejorar la técnica, también desde una única trayectoria inicial, se emplea esta herramienta pero de forma más profunda y se obtienen 122232, 355584, 358457 y 684500 imágenes para la tarea de entrenamiento. Los resultados de esta comparación se pueden revisar en la subsección 5.4.3.

La *fase de test* se realiza con 1483 imágenes distribuidas por todo el entorno y tomadas con las tres iluminaciones disponibles, las imágenes para la *fase de test* se sacan de las bases de datos *COLD-Freiburg Part A Path 2 Cloudy 2, Freiburg Part A Path 2 Night 2 and Freiburg Part A Path 2 Sunny 2*. Todas estas imágenes test se van a evaluar frente a un modelo creado con 556 imágenes diferentes distribuidas por el entorno y tomadas únicamente con iluminación nublado.

5.4. Experimentos

En esta sección se presentan los experimentos y sus resultados. Se muestra el banco de experimentos que se propone para probar la utilidad de la red neuronal siamesa para crear descriptores globales y solucionar el problema de Room Retrieval (recuperación de habitaciones) y la tarea de localización absoluta. Los problemas que se van a experimentar se explican en la sección 5.3.3. La estructura de esta sección es presentar los resultados de estimación de la habitación a la que pertenece (subsección 5.4.1), los obtenidos tras realizar la localización jerárquica (subsección 5.4.2) y por último los resultados de las diferentes variaciones en la experimentación de la localización absoluta (subsección 5.4.3)

5.4.1. Tarea de Room Retrieval

El primero de los problemas evaluados consiste en estimar la habitación en la que se ha capturado la imagen de test. Este problema se conoce como 'room retrieval'. El primer paso es entrenar la red con pares de imágenes tomadas en la misma estancia y en estancias separadas. Después de esto, la efectividad de las redes neuronales siamesas se calcula tras comparar nuevos pares de imágenes y comprobando la etiqueta resultante.

Por tanto, en esta subsección, se estudian los resultados de averiguar si dos imágenes han sido tomadas en la misma estancia o no. Con ello, se puede estimar la habitación en la que una nueva imagen es capturada. Para mayor detalle del proceso se puede consultar la subsección 5.3.5.

Los resultados se expresan en porcentaje de aciertos. Para calcular la configuración más adecuada se evalúan en este punto diferentes características. Por tanto, se evalúan diferentes estructuras para las fases de *extracción de características*, *escalado* y *comparación*. Además, se evalúan otros parámetros como el numero de épocas o el porcentaje de pares similares/diferentes en el entrenamiento.

Como parámetros comunes se entrena la red haciendo uso del optimizador 'Stochastic Gradient Descent (sgd)' con ratio de aprendizaje 0.001 y 'momentum' 0.9.

Elección de las capas de extracción de características:

Durante esta parte se evalúa la primera parte de la red siamesa. La extracción de características se calcula por ramas separadas (cada imagen se trabaja por una rama). Cada rama tiene una red independiente, pero coinciden en arquitectura y tienen los mismos pesos. Las redes utilizadas se pueden revisar en la tabla 5.1 (las capas de activación ReLU no se muestran por brevedad, pero se han utilizado detrás de cada capa conv2d). Los experimentos se han realizado con un entrenamiento en el que se ha utilizado 'tamaño de batch' 256, 5 épocas, 50 % de pares de imágenes capturadas en la misma estancia (imágenes similares) y 50 % de imágenes capturadas en estancias diferentes (imágenes diferentes). La *fase de escalado* se ha realizado utilizando 3 capas totalmente conectadas compuestas por 500-500-5 neuronas.

La precisión se representa en la tabla 5.7 donde se revisan los resultados de precisión obtenidos tras el entrenamiento y test. Adicionalmente,se muestran los porcentajes de aciertos distinguiendo predicciones en la misma habitación y en distintas. La tabla muestra que las redes que ofrecen mejor resultado son la VGG13 y VGG16. Ofrecen sus mejores predicciones cuando tienen que detectar pares de imágenes capturados en la misma habitación (99.44 % y 99.47 % respectivamente) pero también son capaces de detectar cuando dos imágenes han sido tomadas en estancias diferentes con un acierto del 79.86 % and 78.91 %. Además, se puede comprobar como redes simples construidas con un conjunto de tres capas convolucionales son capaces de obtener resultados considerablemente buenos. Estas redes se han llamado 'Simple 1' y 'Simple 2'. Estas capas tienden a sobreentrear, pero a pesar de ello, se obtiene una precisión en el resultado del 97.52 % y 98.20 %. Por último, se observa que los resultados son considerablemente mejores cuando se predice si las imágenes están en la misma habitación que cuando predice que dos imágenes están tomadas en diferentes habitaciones (donde las predicciones no superan en 80 %). Por esta razón, en los siguientes puntos de esta sección se evalúa el entrenamiento de la red haciendo uso de otros parámetros.

Cuadro 5.7: Porcentaje de acierto utilizando diferentes configuraciones para la extracción de características.

Red	Entrenamiento global	Test global	Test misma estancia	Test diferente estancia
Simple 1	97.52 %	84.59 %	98.16 %	**71.03 %**
Simple 2	98.20 %	86.45 %	98.87 %	74.06 %
Alexnet	87.62 %	86.10 %	98.78 %	73.41 %
Densenet	92.12 %	86.06 %	97.61 %	74.52 %
VGG11	92.55 %	87.43 %	99.08 %	75.78 %
VGG11bn	93.10 %	87.51 %	97.49 %	77.53 %
VGG13	94.66 %	**89.65 %**	99.44 %	**79.86 %**
VGG13bn	91.61 %	88.52 %	98.26 %	78.77 %
VGG16	93.30 %	89.19 %	**99.47 %**	78.91 %
VGG16bn	90.90 %	**82.04 %**	**92.68 %**	73.39 %
VGG19	94.42 %	89.17 %	99.30 %	79.04 %
VGG19bn	92.03 %	86.58 %	95.52 %	77.64 %

Porcentaje de imágenes, tamaño de lote y número de épocas:

En este apartado, se evalúan diferentes parámetros que tienen trascendencia en la etapa de entrenamiento. Como se ha explicado en el apartado anterior, el número de imágenes puede modificar notoriamente los resultados. Por ello, se ha comprobado diferentes variaciones cambiando el porcentaje de imágenes en la fase de entrenamiento, el número de veces que pasa cada imagen (especificado con el número de épocas) y el momento en el que se actualizan los datos en la red (dependiente del tamaño de lote). Para reducir la cantidad de información mostrada, se muestran únicamente los resultados significativos, aquellos obtenidos con la redes VGG13, VGG16 y AlexNet. En un primer estudio se muestran los resultados obtenidos al variar porcentaje de imágenes y épocas. Como parámetros comunes se tiene un tamaño de lote de 256 y la fase de escalado se compone de tres capas totalmente conectadas compuestas por 500, 500 y 5 neuronas.

Los resultados se pueden revisar en las tablas 5.8, 5.9 y 5.10. Por un lado, estas tablas muestran la precisión obtenida tras variar el porcentaje de imágenes iguales-diferentes. Cuando aumenta la cantidad de imágenes iguales la precisión de detectar imágenes captadas en la misma habitación aumenta mientras que disminuye la de detectar imágenes obtenidas en distintas estancias; por el contrario, si entrenamos la red con una mayor proporción de imágenes captadas en diferentes estancias, aumenta la posibilidad de detectar esta característica aunque se falla más a la hora de detectar pares de imágenes tomadas en la misma habitación. Los resultados hacen ver que depende más de la red con la que se entrene la captación de características que de los parámetros de entrenamiento. Con todo, parece que es óptimo tener entre un 10 % y un 20 % de pares de imágenes tomadas en la misma estancia. Se recomienda que, durante la etapa de entrenamiento, para la tarea de estimación de estancia el porcentaje de imágenes tomadas en la misma y distinta estancia sea aproximadamente 10 %-90 %

Cuadro 5.8: Porcentaje de acierto utilizando **VGG13** y diferente **número y porcentaje de cada tipo de imágenes** en la fase de entrenamiento

Épocas	Porcentaje (Ig.-Dif.)	Im. entr. (Ig.-Dif.)	Precisión global	Precisión misma est.	Precisión diferente est.
7	5 %-95 %	3046-57882	89.88 %	92.03 %	87.73 %
9	5 %-95 %	3917-74419	91.89 %	92.51 %	91.27 %
11	5 %-95 %	4787-90957	92.20 %	92,71 %	91,70 %
7	10 %-90 %	6093-54835	92.72 %	98.13 %	87.30 %
9	10 %-90 %	7834-70502	94.76 %	98.69 %	90.82 %
11	10 %-90 %	9574-86170	95.08 %	98.90 %	91.25 %
7	25 %-75 %	15232-45696	93.10 %	99.09 %	87,12 %
9	25 %-75 %	19584-58752	93.46 %	99.06 %	87.86 %
11	25 %-75 %	23936-71808	93.53 %	99.21 %	87.85 %

Cuadro 5.9: Porcentaje de acierto utilizando **VGG16** y diferente **número y porcentaje de cada tipo de imágenes** en la fase de entrenamiento

Épocas	Porcentaje (Ig.-Dif.)	Im. entr. (Ig.-Dif.)	Precisión global	Precisión misma est.	Precisión diferente est.
7	5 %-95 %	3046-57882	94.35 %	96.47 %	92.23 %
9	5 %-95 %	3917-74419	94.94 %	96.48 %	93.39 %
11	5 %-95 %	4787-90957	94.24 %	97.77 %	90.72 %
7	10 %-90 %	6093-54835	93.04 %	97.16 %	88.92 %
9	10 %-90 %	7834-70502	94.26 %	97.18 %	91.35 %
11	10 %-90 %	9574-86170	93.59 %	97.96 %	89.22 %
7	25 %-75 %	15232-45696	92.46 %	99.21 %	85.71 %
9	25 %-75 %	19584-58752	92.28 %	99.30 %	85.25 %
11	25 %-75 %	23936-71808	91.78 %	98.81 %	84.74 %
7	40 %-60 %	24371-36557	92.95 %	99.38 %	86.52 %
9	40 %-60 %	31334-47002	92.72 %	99.48 %	85.95 %
11	40 %-60 %	38298-57446	93.28 %	99.50 %	87.05 %

/ 20 %-80 %. Respecto al número de épocas, como el proceso no llega a sobre-entrenar, parece ser que cuanto mayor es el número, mejores son los resultados, aunque más lento es el entrenamiento. Se recomienda un valor de en torno a 10 u 11 épocas.

Hasta el momento, todos los experimentos se han realizado utilizando un tamaño de lote de 256. Para llegar a una configuración optimizada se comprueban otros valores. Las tablas 5.11 y 5.12 muestran la precisión utilizando un tamaño alto de lote (256) y un tamaño bajo (16). Aunque los resultados son parejos, y otra vez dependen de la red que se tiene en la fase de extracción de características, muestran que la precisión global aumenta cuando el tamaño de lote es bajo.

Como conclusión global, tras realizar diferentes entrenamientos en los que se ha variado el porcentaje de imágenes, el tamaño de lote y el número de épocas,

Cuadro 5.10: Porcentaje de acierto utilizando **AlexNet** y diferente **número y porcentaje de cada tipo de imágenes** en la fase de entrenamiento

Épocas	Porcentaje (Ig.-Dif.)	Im. entr. (Ig.-Dif.)	Precisión global	Precisión misma est.	Precisión diferente est.
7	5 %-95 %	3046-57882	92.36 %	90.11 %	94.60 %
11	5 %-95 %	4787-90957	93.58 %	94.08 %	93.07 %
14	5 %-95 %	6093-115763	93.68 %	94.14 %	93.22 %
7	10 %-90 %	6093-54835	92.05 %	94.65 %	89.44 %
11	10 %-90 %	9574-86170	93.41 %	96.84 %	89.98 %
14	10 %-90 %	12186-109670	93.01 %	97.19 %	88.82 %
7	25 %-75 %	15232-45696	90.91	97.54 %	84.28 %
11	25 %-75 %	23936-71808	91.16 %	98.92 %	83.39 %
14	25 %-75 %	30464-91392	90.59 %	98.28 %	82.19 %
7	40 %-60 %	24371-36557	88.33 %	98.80 %	77.85 %
11	40 %-60 %	38298-57446	88.65 %	99.07 %	78.23 %
14	40 %-60 %	48742-73114	88.54 %	99.25 %	77.82 %

Cuadro 5.11: Porcentaje de acierto utilizando **VGG16** y diferente **tamaño de lote** en la fase de entrenamiento.

Tamaño de lote	Épocas	Porcentaje (Ig.-Dif.)	Precisión global	Precisión misma est.	Precisión diferente est.
256	7	5 %-95 %	94.35 %	96.47 %	92.23 %
256	11	5 %-95 %	94.24 %	97.77 %	90.72 %
256	7	10 %-90 %	93.04 %	97.16 %	88.92 %
256	11	10 %-90 %	93.59 %	97.96 %	89.22 %
256	7	25 %-75 %	92.46 %	99.21 %	85.71 %
256	11	25 %-75 %	91.78 %	98.81 %	84.74 %
16	7	5 %-95 %	95.50 %	98.26 %	92.74 %
16	11	5 %-95 %	93.84 %	98.83 %	88.85 %
16	7	10 %-90 %	93.77 %	98.13 %	89.41 %
16	11	10 %-90 %	94.42 %	98.80 %	90.05 %
16	7	25 %-75 %	94.77 %	99.15 %	90.39 %
16	11	25 %-75 %	94.08 %	99.15 %	89.00 %

Cuadro 5.12: Porcentaje de acierto utilizando **AlexNet** y diferente **tamaño de lote** en la fase de entrenamiento.

Tamaño de lote	Épocas	Porcentaje (Ig.-Dif.)	Precisión global	Precisión misma est.	Precisión diferente est.
256	7	5 %-95 %	92.36 %	90.11 %	94.60 %
256	11	5 %-95 %	93.58 %	94.08 %	93.07 %
256	7	10 %-90 %	93.77 %	98.13 %	89.41 %
256	11	10 %-90 %	94.42 %	98.80 %	90.05 %
256	7	25 %-75 %	90.91 %	97.54 %	84.28 %
256	11	25 %-75 %	91.16 %	98.92 %	83.39 %
16	7	5 %-95 %	94.64 %	96.25 %	93.02 %
16	11	5 %-95 %	95.24 %	98.25 %	92.24 %
16	7	10 %-90 %	95.06 %	98.87 %	91.25 %
16	11	10 %-90 %	95.07 %	98.92 %	91.22 %
16	7	25 %-75 %	94.76 %	99.10 %	90.42 %
16	11	25 %-75 %	94.60 %	99.26 %	89.94 %

se puede observar que cuando se aumenta la precisión de acierto para detectar pares de imágenes captadas en la misma estancia, disminuye la precisión de detectar que las imágenes están tomadas en salas diferentes y viceversa. Este error no se produce debido a la red, sino que puede venir producido por la función de pérdida. Esta situación ha sido estudiada por Y. Sun et al. [165] y concluye que utilizar la función *Contrastive Loss* tiene asociada una pérdida de flexibilidad para la optimización, esto significa que al utilizar la función de pérdida *Contrastive Loss* un aumento en la similitud dentro de la misma clase lleva asociado una reducción en la similitud entre clases. Esto quiere decir que si aumenta la precisión de detectar pares de la misma estancia baja la precisión de detectar pares de diferentes estancias y viceversa. El trabajo de Y. Sun et al. propone utilizar la función *Circle Loss* en estos casos. Se propone como trabajo futuro realizar este estudio utilizándola.

Elección de capas para la fase de escalado:

Como se ha explicado en la subsección 5.3.1, la *fase de extracción de características* tiene como salida una matriz que se escala a un vector en la *fase de escalado*. En este apartado se pretende evaluar diferentes combinaciones de capas con las que se puede obtener el vector final. Estos experimentos se realizan entrenando la red con un 10 % de pares de imágenes tomadas en la misma estancia y un 90 % de pares de imágenes tomas en diferentes estancias. Los entrenamientos se realizan con tamaño de lote 16 y usando 7, 11 y 14 épocas.

Las capas de la fase de escalado son capas totalmente conectadas (*fully connected layers*)que se encargan de la tarea de redimensión. Al parecer, las tareas de localización evaluadas no necesitan procesar un gran vector de características, sino que se necesita obtener un vector que enfatice similitudes entre pares de la misma clase y las diferencias de los pares de diferentes clases. Las tablas 5.13

Cuadro 5.13: Porcentaje de acierto utilizando **VGG16** y diferentes **capas en la fase de escalado.**

Capas de escalado	Tamaño de lote	Épocas	Precisión global	Precisión misma est.	Precisión diferente est.
500-500-5	16	7	93.77 %	98.13 %	89.41 %
500-500-5	16	11	94.42 %	98.80 %	90.05 %
500-500-5	16	14	94.75 %	99.10 %	90.39 %
500-100-10	16	7	95.76 %	98.92 %	92.60 %
500-100-10	16	11	95.98 %	99.11 %	92.86 %
500-100-10	16	14	95.44 %	99.18 %	91.70 %
1000-1000-10	16	7	96.16 %	98.90 %	93.41 %
1000-1000-10	16	11	95.63 %	99.10 %	92.16 %
1000-1000-10	16	14	95.27 %	99.10 %	91.44 %

Cuadro 5.14: Porcentaje de acierto utilizando **AlexNet** y diferentes **capas en la fase de escalado.**

Capas de escalado	Tamaño de lote	Épocas	Precisión global	Precisión misma est.	Precisión diferente est.
500-500-5	16	7	93.77 %	98.13 %	89.41 %
500-500-5	16	11	94.42 %	98.80 %	90.05 %
500-500-5	16	14	93.84 %	98.68 %	88.99 %
500-100-10	16	7	95.31 %	98.20 %	92.42 %
500-100-10	16	11	95.41 %	98.98 %	91.83 %
500-100-10	16	14	95.10 %	99.06 %	91.15 %
1000-1000-10	16	7	95.36 %	98.72 %	91.99 %
1000-1000-10	16	11	94.66 %	98.59 %	90.74 %
1000-1000-10	16	14	95.28 %	99.12 %	91.43 %

y 5.14 muestran los resultados utilizando 3 capas totalmente conectadas. Las diferentes versiones consideradas se describen en la tabla 5.2.

Las diferencias entre los resultados tras cambiar las capas en la fase de escalado no son significativas. Globalmente, se obtienen los mejores resultados utilizando 3 capas totalmente conectadas con 1000-1000-10 neuronas por capas.

5.4.2. Localización jerárquica

Una vez la tarea de room retrieval se ha estudiado, se ha demostrado una capacidad para estimar la habitación a la que pertenece una nueva imagen con más del 96 % de acierto. A continuación, se lleva a cabo el estudio de localización jerárquica. Una vez seleccionada la estancia, se realiza una tarea de localización únicamente con las imágenes del modelo almacenadas en la estancia elegida.

En esta sección se evalúa el error una vez se ha realizado la tarea de elección de estancia y se trata de estudiar la pose concreta dentro de la estancia. La tabla

Cuadro 5.15: Error en la tarea de localización usando **VGG16-500-500-5, 30 épocas, 16 de tamaño de lote** en las diferentes estancias.

Estancias	Error en nublado	Error en noche	Error en soleado
1P0-A	0.0312 m	0.2434 m	0.2937 m
2P01-A	0.0653 m	0.2345 m	0.5137 m
2P02-A	0.0363 m	0.2897 m	0.2897 m
CR-A	0.0436 m	0.4140 m	0.7903 m
KT-A	∼0 m	0.2704 m	0.3027 m
LO-A	0.0839 m	0.3686 m	0.3710 m
PA-A	∼0 m	0.2821 m	0.2522 m
ST-A	0.0766 m	0.1981 m	0.3262 m
TL-A	∼0 m	0.1737 m	0.2145 m

5.15 muestra los resultados obtenidos utilizando la red VGG16 preentrenada, 3 capas totalmente conectadas de 500-500-5 neuronas en la fase de escalado, 30 épocas de entrenamiento, 16 de tamaño de lote y el optimizador sgd. Se muestra el error de localización en tres estados de iluminación posible (nublado, noche y soleado).

Las estancias han sido presentadas previamente en la tabla 5.5. En esta tabla se muestra la máxima distancia posible entre imágenes tomadas en la misma estancia.

Esta información ayuda a comprender el buen funcionamiento de las redes siamesas ya que el error obtenido es relativamente pequeño. El error tras realizar localización jerárquica en un dia nublado no llega a los 10 cm en ninguno de los casos. En las opciones de localización de noche los errores medios están por debajo de los 30 cm, excepto en el pasillo y una de las oficinas. Por último, con el entorno iluminado por mucha luz natural el error también es relativamente bajo excepto por un aumento a un error medio de 80 cm en el pasillo y 51 cm en uno de los despachos.

5.4.3. Estimación de la posición. Localización absoluta

La tarea de localización absoluta se realiza mediante comparación de pares de imágenes (image retrieval). La imagen que se pretende localizar se compara con todas las imágenes del modelo y aquella con mayor valor de similitud se selecciona como imagen más cercana y se comprueba el error. Para más detalles se puede comprobar la subsección 5.3.5.

Para evaluar esta tarea se han realizado diferentes experimentos, los cuales permiten elegir la mejor configuración. Como en todas las tareas previas, utilizar la red preentrenada VGG16 para la tarea de extracción de características ofrece los mejores resultados. Se ha utilizado esta red para solucionar el problema de localización absoluta con ella y obtener la mejor configuración del resto de parámetros.

Como parámetros comunes se ha entrenado la red utilizado un tamaño de lote de 16, el optimizador sgd (*Stochastic Gradient Descent*) un ratio de aprendizaje de

Cuadro 5.16: Error en la tarea de localización absoluta variando la **fase de escalado**.

Fase de caract.	Fase de escalado	Épocas	Porcentaje (Ig.-Dif.)	Error global
VGG16	500-500-5	30	50 %-50 %	0.5821 m
VGG16	1000-1000-10	30	50 %-50 %	0.5904 m
VGG16	4096-4096-1000	30	50 %-50 %	0.8313 m

0.001 y un momento de 0.9. Adicionalmente, el número de épocas se ha sintonizado a 30 épocas, ya que en este caso son más las características a aprender.

A continuación, se evalúan y comentan los diferentes parámetros explicados en el apartado de estimación de estancia (subsección 5.4.1) y que tienen influencia en la tarea de localización.

Elección de capas para la fase de escalado:

La tabla 5.16 muestra una comparación de los resultados tras evaluar diferentes capas en la fase de escalado. Como recordatorio, la fase de escalado se encarga de adaptar el tamaño del descriptor, de la matriz obtenida de la fase de extracción de características a los vectores que finalmente se comparan. Los resultados muestran que vectores pequeños ofrecen mejores comparaciones que descriptores más largos en la tarea de localización absoluta. El entrenamiento de una red siamesa puede ofrecer una tarea de localización con un error de 0.5821m cuando se utiliza tres capas totalmente conectadas con 500, 500 y 5 neuronas.

Porcentaje de imágenes:

La tabla 5.17 muestra una comparativa al variar el la proporción de pares de imágenes iguales/diferentes. Una vez seleccionado el mejor resultado hasta el momento (el cual es usando la red VGG16 en la fase de extracción de características, tres capas totalmente conectadas con 500, 500 y 5 neuronas en la fase de escalado y 30 épocas) se varía el porcentaje de pares de imágenes iguales y diferentes que se estudia en el entrenamiento.

El resultado muestra que el error más bajo se obtiene entrenando la red con una proporción de 40 % de parejas de imágenes tomadas en la misma estancia y un 60 % de pares de imágenes tomadas en estancias diferentes. Tras estudiar los resultados, se concluye que un alto porcentaje de pares tomados en la misma estancia (y un bajo porcentaje tomada en estancias diferentes) no ofrece resultados óptimos, pero un porcentaje pequeño de pares tomados en la misma estancia tampoco ofrece cantidad importante de información para aprender características y los resultados no son buenos. Por todo ello, hasta el momento la proporción que mejor funciona es 40 %-60 %.

Cuadro 5.17: Error en la tarea de localización absoluta variando el **porcentaje de imágenes**.

Fase de caract.	Fase de escalado	Porcentaje (Ig.-Dif.)	Error global	Error nublado	Error noche	Error soleado
VGG16	500-500-5	80 %-20 %	*0.6284 m*	0.1826 m	0.5600 m	0.8023 m
VGG16	500-500-5	70 %-30 %	*0.6035 m*	0.1753 m	0.5375 m	0.7706 m
VGG16	500-500-5	60 %-40 %	*0.6006 m*	0.1802 m	0.4991 m	0.8649 m
VGG16	500-500-5	50 %-50 %	*0.5821 m*	0.1747 m	0.4837 m	0.8383 m
VGG16	500-500-5	40 %-60 %	*0.5097 m*	0.1481 m	0.4547 m	0.6508 m
VGG16	500-500-5	30 %-70 %	*0.5193 m*	0.1518 m	0.4908 m	0.6630 m
VGG16	500-500-5	20 %-80 %	*0.5202 m*	0.1521 m	0.4917 m	0.6642 m

Data Augmentation (DA):

Hasta este momento todo el entrenamiento se ha realizado haciendo uso de las imágenes contenidas en la base de datos inicial. Llegados a este punto se pretende hacer uso de la herramienta de *data augmentation*. Como se ha visto en la subsección 5.3.2, la técnica de *data augmentation* permite construir artificialmente más imágenes con las que llevar a cabo un mejor entrenamiento. Como se ha visto en la literatura [26, 152], esta técnica puede mejorar el rendimiento de redes simples sin necesidad de tener más información de inicio, simplemente creando más imágenes a partir de las que se tiene. Artificialmente se añaden efectos que pueden aparecer en condiciones reales y se entrena a la red utilizando estos efectos.

La técnica de *data augmentation* se aplica a las imágenes de la base que se han tomado el día nublado. Y depende de cuánto de profunda se haga la técnica, y por tanto de la cantidad de imágenes diferentes con las que se entrena la red, se obtendrá un mayor número de imágenes. En la tabla 5.18 se pueden comprobar los resultados tras aplicar 5 tipos diferentes de 'data augmentation' y entrenar la red con 4448 imágenes diferentes a 684500 imágenes diferentes. Por comparación, las dos primeras filas muestran los resultados tras entrenar la red sin aplicar *data augmentation*. Cuando se entrena la red sin *data augmentation* la red entrena combinando imágenes de nublado, soleado y noche mientras que al realizar el entrenamiento con *data augmentation* este se aplica únicamente a imágenes de nublado. Esta es la explicación que justifica un mejor resultado en soleado cuando no se aplica la técnica de *data augmentation*.

Evaluando la tabla se comprueba que cuanto mayor sea la generación de imágenes, cuanto más profundo es el DA, mejores resultados se obtienen. Esto se hace notorio cuando se localiza en días soleados o de noche. Esto permite comprobar que el funcionamiento del *data augmentation* permite mejorar los resultados de localización. Asimismo, también se comprueba que una proporción 40 %-60 % es más recomendable que realizar el entrenamiento con pares de imágenes iguales-diferentes 50 %-50 %.

Comparación general de métodos:

Cuadro 5.18: Error en la tarea de localización absoluta usando **VGG16** y **Data Augmentation**.

Fase de escalado	Cantidad de imágenes	Porcentaje (Ig.-Dif.)	Error nublado	Error noche	Error soleado
500-500-5	*8486 (Sin DA)*	*50 %-50 %*	*0.1747 m*	*0.4837 m*	*0.8383 m*
500-500-5	*8486 (Sin DA)*	*40 %-60 %*	*0.1481 m*	*0.4547 m*	*0.6508 m*
500-500-5	4448	50 %-50 %	0.0630 m	0.4682 m	1.1880 m
500-500-5	4448	40 %-60 %	0.0608 m	0.4682 m	1.2703 m
500-500-5	122232	50 %-50 %	0.0375 m	0.3961 m	1.8257 m
500-500-5	122232	40 %-60 %	0.0407 m	0.4028 m	1.3733 m
500-500-5	355584	50 %-50 %	0.0440 m	0.3591 m	1.2264 m
500-500-5	355584	40 %-60 %	0.0418 m	0.3176 m	1.0050 m
500-500-5	358548	50 %-50 %	0.0395 m	0.7547 m	1.2825 m
500-500-5	358548	40 %-60 %	0.0375 m	0.2675 m	1.051 m
500-500-5	684500	50 %-50 %	0.0342 m	0.2909 m	1.2097 m
500-500-5	684500	40 %-60 %	0.0325 m	0.2573 m	0.9913 m

Finalmente, las redes siamesas se han comparado con otros descriptores de apariencia global para la tareas de localización absoluta en la trayectoria de *Freiburg* [135]. Por un lado, se evalúan otras técnicas de *deep learning*, para ello se compara con la red simple (una sola rama de entrada) Alexnet utilizada para la misma tarea por Cebollada et al. [34]. Por otro lado, se evalúa también con las técnicas clásicas de descriptores analíticos [146] y descritas en el capítulo de este trabajo 3.

La tabla 5.19 muestra los resultados de esta comparación. En ella se pueden observar los diferentes descriptores de apariencia global evaluados y el mejor error de localización absoluta que proporcionan. Asimismo, también se muestra el tamaño de descriptor que se consigue. Como recordatorio, todas las técnicas tratan de localizar el robot móvil en diferentes condiciones de iluminación sobre un mapa construido en un entorno de nublado. Cuando se entrenan las redes neuronales,se hace con las imágenes de todos los entornos, mientras que cuando se aplica la técnica de DA esta se hace únicamente con las imágenes de nublado. Todas las localizaciones se realizan con información visual y no utilizando otro tipo de sensores ni datos métricos.

Esta tabla 5.19 muestra que las técnicas de descriptores analíticos son capaces de localizar mejor el robot cuando se estudia su posición con las misma condiciones de iluminación con las que se ha construido el modelo, sin embargo, cuando se quiere localizar en entornos con una iluminación diferente el error aumenta notoriamente, especialmente en entornos soleados. Además, al comparar las redes neuronales, los resultados son parejos. Las redes siamesas funcionan mejor en entorno nublado mientras que las redes CNN simple lo hacen mejor en soleado o de noche. Se puede destacar la red siamesa lo hace con descriptores mucho menores, por lo que se puede concluir que condensan muy bien la información.

Cuadro 5.19: Evaluación de métodos para la tarea de localización absoluta.

Técnica de descriptor global	Error nublado	Error noche	Error soleado	Tamaño del descriptor
CNN simple				
-Alexnet	0.29 m	0.29 m	0.69 m	4096
-Alexnet + DA	0.25 m	0.24 m	0.87 m	4096
Red siamesa simple				
-Red siamesa	0.1481 m	0.4547 m	0.6508 m	5
-Red siamesa + DA	0.0325 m	0.2573 m	0.9913 m	5
Descriptores analíticos				
-FS	0.0566 m	0.7330 m	3.5053 m	32
-HOG	0.0581 m	0.1931 m	1.1686 m	256
-Gist	0.0511 m	0.2355 m	1.6888 m	512
-WS	0.0811 m	0.3021 m	0.8425 m	131072
-BG	0.0526 m	1.1262 m	1.6369 m	1024

5.5. Conclusión

La finalidad de este capítulo se ha centrado en la resolución de la tarea de localización haciendo uso del *deep learning* y las redes neuronales, en especial en el estudio de las redes siamesas. Se ha realizado una evaluación de estas para la resolución de tres supuestos: la estimación de habitación (tarea de room retrieval), la localización jerárquica y la estimación absoluta de la posición. En primer lugar, en la tarea de room retrieval se ha tratado de conocer si dos imágenes han sido capturadas en la misma o en diferente habitación. En segundo lugar, en la tarea de localización jerárquica el robot trata de ubicarse dentro de una estancia concreta, en este caso el robot solo comparará la nueva imagen con las imágenes almacenadas en el modelo de la estancia estimada. Por último, en la tarea de localización absoluta la nueva imagen se compara con todas las imágenes del modelo.

Para la obtención de información del entorno se ha utilizado un sensor de visión catadióptico montado sobre un robot móvil y se ha tratado de estimar habitación y posición haciendo uso únicamente de información visual. Para extraer información relevante de las imágenes se hace uso de las redes siamesas. Estas redes tienen la particularidad de comparar dos imágenes de entrada, de cada imagen se obtiene un vector global que define la imagen y estos se comparan dando un porcentaje de similitud entre imágenes. Son varios parámetros de las redes siamesas los que se han evaluado: red de extracción de características, red de la fase de escalado, tamaño de lote, el número de epoch o épocas, el porcentaje de imágenes similares/diferentes con los que se entrena la red, tasa de aprendizaje, optimizador... Además, también se han estudiado técnicas de *data augmentation* para generar imágenes artificiales con las que entrenar la red. Diferentes versiones de *data augmentation* se han evaluado durante el trabajo.

La principal aportación de este capítulo es la demostración de que se pueden utilizar las redes siamesas para llevar a cabo tareas de localización. Una de la etapas

clave es la construcción de estas redes para que tengan dos ramas de entrada. Además, es crucial la elección de capas para la fase de extracción de características. Para esta primera fase se ha realizado un estudio exhaustivo de diferentes configuraciones entre las que se destaca la adaptación, mediante *transfer learning*, de redes suficientemente contrastadas como son Alexnet [81], DenseNet [61], VGG11, VGG13, VGG16 o VGG19 [158], además se han evaluado dos configuraciones simples de nueva creación a las que hemos llamado Simple 1 y Simple 2.

De entre las opciones para la fase de extracción de características destacan los resultados en la tarea de **estimación de la estancia** obtenidos por VGG13 y VGG16, en la comparación para detectar si dos imágenes pertenecen a la misma estancia ofrecen un acierto del 99.44 % y 99.47 % mientras que su porcentaje de acierto al detectar que las imágenes son de diferente estancia es de 79.86 % y 78.91 %. Se considera unos resultados adecuados ya que la principal finalidad de esta tarea es la de encontrar la estancia en la que está ubicado el robot. Varias configuraciones aciertan que está en una estancia con más de un 98 % de acierto, mientras que al determinar que son imágenes captadas en estancias separadas tiene una seguridad del 75 %. Tener un acierto del 98 % al estimar que dos imágenes pertenecen a la misma estancia ofrece una gran seguridad, aunque haya un 25 % de las veces que se estime que son imágenes de estancias diferentes y realmente estén en la misma. Se trata de un resultado conservador ya que si se tiene que equivocar lo hará con falsos negativos y casi nunca con falsos positivos.

También se han analizado otros hiperparámetros para estas tareas, por ejemplo el porcentaje de imágenes misma estancia/diferente estancia, el numero de épocas o el tamaño de lote. Este ajuste ha logrado un aumento en el porcentaje de acierto cuando detecta que dos estancias son diferentes manteniendo los altos porcentajes de acierto cuando se detecta que dos imágenes son de la misma estancia, esto provoca que la precisión global aumente. Mientras se obtenía un acierto global de 89.65 % para VGG13, 89.19 % para VGG19 o 86.10 % para AlexNet sin ajustar estos parámetros, tras ello se obtiene un 95.08 %, 95.50 % y 95.24 % de precisión global respectivamente. De este estudio se deduce que para la tarea de **estimación de la estancia** es más adecuado tener porcentajes de imágenes en el entrenamiento en torno al 5 %-95 % o 10 %-90 % de imágenes captadas en la misma estancia respecto a las captadas en estancias diferentes, un numero de épocas en torno a 10 y un número bajo de tamaño de lote (en torno a 16). Por último, también se han evaluado las diferentes configuraciones para la fase de escalado. Se ha detectado que la modificación de estas capas no hace variar demasiado los resultados. Con todo, construcciones de tres capas totalmente conectadas con 1000, 1000 y 10 neuronas ofrecen los mejores resultados globales y una configuración de 500, 100 y 10 neuronas ofrece unos resultados de acierto cuando se detecta la misma estancia superiores al 99 % (aunque baja el porcentaje de estimación de diferentes estancias).

En segundo lugar, se ha realizado un estudio para determinar la posición exacta dentro de una estancia concreta. Esta tarea se ha presentado como la tarea de **localización jerárquica** y se ha evaluado ante diferentes características de iluminación. Los

experimentos muestran un error de localización inferior a los 0.07 m. cuando se evalúa el en torno durante un día nublado de baja iluminación natural. Los errores están entorno a los 0.20 m cuando se evalúa por la noche y con alto nivel de luz artificial, exceptuando la estancia del pasillo en la que el error sube a 0.41 m. Por último, el error de localización ante un día soleado con alta iluminación natural está entre los 0.20 m y los 0.32 m, exceptuando nuevamente el pasillo que supone un error medio de 0.79 m y en uno de los despachos donde el error medio asciende a 0.51 m. Se consideran unos errores de localización adecuados debido a la complejidad de la tarea.

También se han analizado la tarea de **localización absoluta** en la que la nueva imagen test se evalúa con todas las imágenes del modelo para encontrar su posición. En esta tarea se ha evaluado diferentes parámetros. Por un lado, se detecta que en la fase de escalado la mejor opción es tener 3 capas con 500, 500 y 5 neuronas totalmente conectadas. Respecto al número de imágenes la mejor opción es entrenar la red con un 40 % de pares de imágenes de la misma estancia y un 60 % de pares de imágenes tomadas en estancias separadas. Con esos parámetros y tras entrenar la red con 30 épocas se obtiene un error medio de localización de 0.15 m en entorno nublado, 0.45 m en entorno de noche y de 0.65 m en entorno soleado.

Por último, se ha hecho uso del 'data augmentation' para obtener un mayor número de imágenes para el entrenamiento. Cuanto mayor es el número de imágenes con las que se entrena, mejores eran los resultados que se han obtenido, y al tratarse de imágenes diferentes las redes no han llegado a sobreentrenar. Tras realizar un 'data augmentation' exhaustivo y obtener hasta 684500 imágenes se logra tener un error medio de localización absoluta de 0.03 m para entorno nublado, 0.26 m con entorno de noche y de 0.99 m con día soleado.

Con el estudio realizado durante este capítulo se demuestra la utilidad de las redes siamesas para comparar imágenes y dar un valor de similitud entre ambas. Esta alternativa se usa para realizar una tarea de localización de robots móviles utilizando información visual y se experimenta su uso para realizar las tareas de estimación de la estancia, de localización jerárquica y de localización absoluta con unos resultados satisfactorios.

6 Conclusiones y Trabajos Futuros

Tras presentar en detalle todo el trabajo de investigación realizado en el marco de la presente tesis, este capítulo resume las principales aportaciones de los estudios realizados. Además, el apartado 6.2 introduce posibles ampliaciones y trabajos futuros que se pueden realizar a partir de las líneas de investigación presentadas en esta tesis.

6.1. Contribuciones

Esta tesis ha presentado un análisis de los principales descriptores visuales de apariencia global para tareas de localización, un nuevo método para poder realizar la tarea de creación de mapas de manera incremental en entornos de interior y, por último, se ha explorado el uso de redes siamesas, que utilizan herramientas de 'deep learning' para comparar simultáneamente pares de imágenes. El análisis de los estudios ha consistido en evaluar la eficiencia de los métodos propuestos mediante la medición del error de localización medio, medidas de porcentaje de acierto, comparación con otras técnicas de mapeo y el tiempo medio de computación. Estos análisis fueron desarrollados mediante herramientas de simulación: Matlab ® , C++ y Python. Los principales logros y aportaciones de esta tesis se resumen a continuación.

Capítulo 3

- Se han presentado diferentes métodos para compactar la información visual en único vector. La información es adquirida por imágenes omnidirecionales que se transforman a imágenes panorámicas, estas se describen realizando operaciones matemáticas y se obtiene un vector que representa globalmente la escena. Para

realizar el estudio de localización, el robot tiene un modelo del entorno almacenado y es capaz de estimar la posición y orientación de una nueva captura tras buscar el vecino más cercano en el modelo.

- Se han analizan seis familias de descriptores en tareas de localización absoluta, en la que el robot detecta su posición más cercana, calcula el error medio y posteriormente estima su orientación relativa respecto a la imagen más cercana. Adicionalmente, también se estudia el coste computacional del proceso y se incluye la influencia de cada uno de los parámetros que tienen posibilidad de ser modificados durante el proceso de descripción. Los resultados expuestos han demostrado que los métodos de apariencia global son un enfoque viable para llevar a cabo la tarea de localización.

- Durante el proceso de análisis se ha estudiado la influencia de distintos efectos negativos que pueden ocurrir en tareas de localización reales. En concreto, se han estudiado efectos adversos como la aparición de ruido en las imágenes o de oclusiones que no permitan ver por completo la escena. En el capítulo 3 se expone la comparativa de realizar la tarea de localización en condiciones ideales y cuando estos efectos aparecen en la captura de información. El análisis ha demostrado que ciertos descriptores de apariencia global son una opción fiable para reconocer información de la imagen en conjunto y poder realizar tareas en entornos reales y cambiantes.

- El estudio revela que los algoritmos de FS y RT presentan un coste computacional bajo, igual que ciertas configuraciones específicas de HOG y *gist*, pero los descriptores Wi-SURF y BRIEF-gist son menos competitivos computacionalmente hablando. Desde este punto de vista, FS, RT, HOG y *gist* son más útiles para tareas en tiempo real.

- En lo relativo a la tarea de localización. Los resultados muestran que, a la hora de encontrar el vecino más cercano, los descriptores Wi-SURF y BRIEF-gist presentan un mejor ratio de acierto que el resto de métodos. Los métodos HOG y *gist* también ofrecen unos resultados relativos bastante adecuados con ciertas combinaciones de parámetros, teniendo HOG la mejor relación entre coste computacional y ratio de acierto. Por su lado, los descriptores FS y RT no ofrecen resultados competentes en esta tarea.

- Cuando se ha realizado la tarea de localización absoluta en condiciones ideales y reales, los experimentos muestran diferentes conclusiones. Primero, los descriptores HOG y *gist* presentan un resultado competente bajo condiciones ideales, pero además son bastante robustos ante ruido y oclusiones, en especial el descriptor HOG. Segundo, el descriptor Wi-SURF proporciona los mejores resultados en condiciones ideales pero el efecto que tienen los fenómenos adversos negativos en la imagen influye mucho en la tarea de localización, obteniendo resultados de error poco asumibles. Por último, BRIEF-gist es el descriptor más robusto ante efectos adversos, pero su calidad no es remarcable en condiciones ideales. En resumen, los mejores resultados en condiciones ideales se obtienen usando el

descriptor Wi-SURF, pero ante fenómenos adversos, como podría ser el ruido o las oclusiones es mejor hacer uso de los descriptores HOG y *gist*.

Capítulo 4

- Se ha demostrado la posibilidad de usar las imágenes omnidirecionales y los descriptores de apariencia global para la tarea de creación de mapas. En especial, se propone un algoritmo para crear los mapas de manera incremental mientras el robot continúa visitando el entorno.
- Además, se muestra una comparativa para evaluar los resultados obtenidos por el algoritmo propuesto con un algoritmo de 'clustering' espectral clásico que funciona *off-line* y de manera absoluta (es necesario que todas las imágenes estén disponibles de antemano). Para evaluar cuánto de buenos son los métodos se ha calculado la silueta media. Para calcular la silueta se considera los descriptores de las imágenes como entidades y se estudia cuánto se parece cada uno de ellos a las entidades de su nodo y cuánto de diferente es al resto de los nodos.
- El descriptor HOG demuestra tener mejores resultados, ya que la silueta media obtenida es similar a la silueta obtenida usando un método de clustering espectral clásico, pese que a que el algoritmo propuesto va obteniendo información conforme se visita el entorno. De hecho, si los parámetros se ajustan adecuadamente, puede dar lugar a resultados de silueta muy parejos al clustering espectral.
- Los resultados muestran un deterioro en el rendimiento del algoritmo cuando la base de datos contiene un número excesivo de imágenes o cuando el entorno presenta características complejas (por ejemplo, muchas ventanas o paredes de cristal o características que tiendan al *aliasing visual*).

Capítulo 5

- Se ha presentado un método de localización basado en la creación y entrenamiento de redes neuronales. Se ha experimentado la resolución de tareas de localización haciendo uso del 'deep leaning', en especial, se presenta una técnica poco explorada en robótica móvil, denominada redes siamesas. En este tipo de redes dos imágenes se comparan de manera simultánea y se devuelve un valor de similitud entre imágenes.
- Se han resuelto tres tareas de localización. Primero, la tarea de 'room retieval', que trata de encontrar la estancia más probable en la que el robot está ubicado. Segundo, se ha estudiado la tarea de localización jerárquica en la que primero se encuentra el nodo de pertenencia más probable y de entre las imágenes del nodo se encuentra la posición más probable. Por último, también se resuelve la tarea de localización absoluta en la que una nueva imagen captada se compara con todas las imágenes del modelo y se encuentra la posición más probable, sin

ejecutar un paso previo de localización gruesa. Los estudios que se han realizado demuestran que las redes siamesas son capaces de abordar una tarea de localización con resultados competentes. Además la tarea se resuelve utilizando menos descriptores y de menor tamaño que con otras técnicas. Esto permite realizar un gran número de comparaciones con un menor coste computacional.

- Se ha llevado a cabo un 'data augmentation' extenso, que considera los efectos visuales que pueden ocurrir en entornos y condiciones de trabajo reales. Se ha demostrado que el aumento de imágenes con esta herramienta es una técnica adecuada para ampliar el conjunto de datos original. Esta técnica tiene el objetivo de llevar a cabo el entrenamiento de redes neuronales con la posibilidad de poder anticipar alteraciones en futuras tareas de localización reales.

- Durante el capítulo 5 se presentan los resultados tras realizar un estudio con diferentes combinaciones de redes, entrenamientos e hiperparámetros. Los resultados han demostrado la robustez de los descriptores provenientes de las redes siamesas.

- Por último, se realiza una comparativa para resolver una tarea de localización real entre los descriptores analíticos de descripción global, los descriptores de redes neuronales simples y los que provienen de redes siamesas. Este análisis muestra que en varias condiciones las redes siamesas mejoran a sus predecesoras, aun realizando la comparación con descriptores de menor tamaño.

6.2. Trabajos Futuros

Por último, se proponen algunos trabajos de investigación futuros que pueden derivar de las investigaciones, experimentos y resultados presentados a lo largo de la presente tesis.

- **Desarrollar mejoras en la tarea de mapeo incremental.** La herramienta de creación de mapas de manera incremental desarrollada en este trabajo es susceptible de algunas mejoras. Los experimentos muestran que funciona bien en entornos de tamaño medio, pero cuando la cantidad de imágenes se incrementa notoriamente o existe situaciones bastante desfavorables (como mucha iluminación natural que satura parcialmente las escenas o personas que ocluyen la imagen) la técnica puede fallar. Desarrollar mejoras al método creado debe ser una prioridad, así como el mantenimiento de los mapas creados. También es recomendable seguir estudiando parámetros que influyen en el algoritmo y realizar ajustes para realizar la tarea en entornos desestructurados y de exterior.

- **Desarrollo de un algoritmo SLAM visual basado en los métodos jerárquicos.** Con los trabajos presentados en esta tesis se han validado métodos para realizar tareas de creación de mapas y localización jerárquica con imágenes omnidireccionales. Por lo tanto, se puede desarrollar una herramienta enfocada en la tarea de SLAM basada en los métodos estudiados. El objetivo de este enfoque

es desarrollar un sistema de navegación de robots móviles lo más autónomo posible. Este enfoque puede estar basado en la idea de agrupamiento incremental, tal y como se ha visto en trabajos en este documento, de esta manera se puede abordar la tarea de modelado del entorno y estimación de la pose del robot simultáneamente mientras el robot explora el nuevo entorno.

- **Evaluación de arquitecturas de aprendizaje profundo más novedosas.** Con respecto al uso de Inteligencia Artificial para resolver tareas de navegación, en la presente tesis se han realizado tareas de localización usando CNNs a partir de redes desarrolladas previamente como AlexNet, DenseNet o VGG16. Sin embargo, estas redes fueron creadas en la década de 2010. Por lo tanto, en trabajos futuros, se considerarán arquitecturas de CNNs más recientes como punto de partida para construir nuestra red neuronal. De esta forma, puede ser posible obtener resultados más óptimos en condiciones de trabajo dinámicas y desafiantes, así como un mejor coste computacional.

- **Optimización de la red neuronal siamesa.** Los trabajos desarrollados con este tipo de redes han supuesto un primer análisis de estas soluciones para tareas de localización, que han dado unos resultados relativamente competentes. No obstante, en el futuro se realizará un estudio más exhaustivo de todos los hiperparámetros y combinaciones con el objetivo de optimizar los procesos de extracción de características, escalado y comparación. Otro de los trabajos a explorar sería utilizar otras funciones de pérdida, como *'circle loss function'*, que optimicen el entrenamiento de la red. Además, otra línea de investigación futura sería optimizar el proceso de entrenamiento. Por último, también es preciso abordar un proceso de agrupación en clusters haciendo uso de estas nuevas redes.

- **Desarrollar un estudio exhaustivo de la herramienta de 'Data Augmentation'.** En esta tesis se presentó el 'Data Augmentation' de propia creación el cual permitía generar efectos de distorsiones lumínicas las cuales es interesante considerar en el entrenamiento, ya que pueden surgir en situaciones reales. El estudio concluyó con resultados satisfactorios. Sería interesante crear más formas de generar nuevas imágenes con modificaciones artificiales que permitan aprender durante la etapa de entrenamiento posibles problemas a los que se enfrentará el robot.

- **Desarrollar técnicas de 'transfer learning' para estimar con robustez la posición del robot.** Continuando con el trabajo presentado en la tesis, el siguiente paso será utilizar redes neuronales que permiten realizar un 'transfer learning' con el objetivo de resolver una tarea de regresión. En este caso, la tarea de regresión puede consistir en estimar directamente la posición (x, y) del robot dentro del entorno. De esta manera, se deben resolver varias cuestiones. Por ejemplo, entrenando una CNN con dos salidas, entrenando una CNN para generar la coordenada x y otra para y o utilizar redes siamesas o tripletas para resolver esta tarea.

A Apéndice: Conjunto de publicaciones

Las mayores contribuciones de esta tesis están respaldadas por artículos publicados en revisas indexadas en JCR (Science Edition). Estos artículos están presentados a continuación. The major contributions made in the present thesis are supported by four papers published in journals ranked in JCR (Science Edition). The metadata of these journal papers are presented next:

Artículo 1

The Role of Global Appearance of Omnidirectional Images in Relative Distance and Orientation Retrieval. [146]
V. Román, L. Payá, A. Peidró, M. Ballesta, O. Reinoso
Sensors. 21(10), 3327 (Mayo 2021)
ISSN:1424-8220. Ed. MDPI
JCR-SCI Impact Factor: 3.576, Quartile **Q1**
Web: https://doi.org/10.3390/s21103327
DOI: 10.3390/s21103327

Artículo 2

Creating Incremental Models of Indoor Environments through Omnidirectional Imaging. [142]
V. Román, L. Payá, S. Cebollada, O. Reinoso
Applied Sciences. Vol 10(18),6480 (Septiembre 2020)
ISSN:2076-3417. Ed. MDPI

JCR-SCI Impact Factor: 2.679, Quartile **Q2**
Web: https://doi.org/10.3390/app10186480
DOI: 10.3390/app10186480

Las publicaciones de estos artículos están adjuntos a continuación.

sensors

MDPI

Article

The Role of Global Appearance of Omnidirectional Images in Relative Distance and Orientation Retrieval

Vicente Román *, Luis Payá, Adrián Peidró, Mónica Ballesta and Oscar Reinoso

Department of Systems Engineering and Automation, Miguel Hernández University, 03202 Alicante, Spain; lpaya@umh.es (L.P.); apeidro@umh.es (A.P.); m.ballesta@umh.es (M.B.); o.reinoso@umh.es (O.R.)
* Correspondence: v.roman@umh.es; Tel.: +34-96-665-8859

Abstract: Over the last few years, mobile robotics has experienced a great development thanks to the wide variety of problems that can be solved with this technology. An autonomous mobile robot must be able to operate in a priori unknown environments, planning its trajectory and navigating to the required target points. With this aim, it is crucial solving the mapping and localization problems with accuracy and acceptable computational cost. The use of omnidirectional vision systems has emerged as a robust choice thanks to the big quantity of information they can extract from the environment. The images must be processed to obtain relevant information that permits solving robustly the mapping and localization problems. The classical frameworks to address this problem are based on the extraction, description and tracking of local features or landmarks. However, more recently, a new family of methods has emerged as a robust alternative in mobile robotics. It consists of describing each image as a whole, what leads to conceptually simpler algorithms. While methods based on local features have been extensively studied and compared in the literature, those based on global appearance still merit a deep study to uncover their performance. In this work, a comparative evaluation of six global-appearance description techniques in localization tasks is carried out, both in terms of accuracy and computational cost. Some sets of images captured in a real environment are used with this aim, including some typical phenomena such as changes in lighting conditions, visual aliasing, partial occlusions and noise.

Keywords: omnidirectional imaging; global appearance description; localization; image retrieval; relative orientation; fourier signature; histogram of oriented gradients; gist

Citation: Román, V.; Payá, L.; Peidró, A.; Ballesta, M.; Reinoso, O. The Role of Global Appearance of Omnidirectional Images in Relative Distance and Orientation Retrieval. *Sensors* **2021**, *21*, 3327. https://doi.org/10.3390/s21103327

Academic Editor: Radu Danescu

Received: 5 April 2021
Accepted: 4 May 2021
Published: 11 May 2021

1. Introduction

Nowadays, the presence of mobile robots has increased substantially in many areas, such as industry, households, transportation and education. As their abilities in perception and computation have increased, they have become an efficient tool to perform a wide range of tasks and they are expected to play a crucial role in the development of some activities. In this context, map building and localization are two of the main abilities a robot must develop to be really autonomous. Finding a solution to both problems, balancing accuracy, efficiency and robustness, is very important so that a robot can navigate autonomously and safely through real working environments [1].

In the field of perception, vision sensors have become a widespread tool to get information from the environment [2] due to several factors: the big amount of information they can capture with a relatively low cost; the availability of the data they provide (unlike GPS, whose signal may not be available temporarily, indoors or in narrow outdoor areas); the variety of configurations that they permit, from single-view cameras to binocular or trinocular systems; and the possibility of carrying out other high-level tasks such as people detection. Among the available configurations, catadioptric vision systems stand out thanks to their wide field of view, up to 360 deg around the camera axis [3]. The information captured with these systems can be projected onto varied surfaces, what permits different mathematical approaches depending on the type of task to solve [4]. Omnidirectional

images are particularly effective comparing to conventional images due to the fact that they capture a global context of the environment. Therefore, with this kind of information, global features constitute an effective alternative, compared to local features, to many tasks, such as, for example, the reconstruction of complex indoor environments. In this regard, Sun et al. [5] and Pintone et al. [6] make use of deep learning approaches [7–9] to panoramic image analysis, with the objective of understanding the layout of indoor environments.

Solving the mapping and localization problems using only visual information is challenging. Images are highly dimensional data and they usually contain much redundant information. This information tends to change not only when the robot moves but also under some other usual circumstances such as changes in the external lighting conditions, noise during the acquisition of the image and occlusions due to the presence of, e.g., people in the environment. In addition, when a robot has to operate in indoor environments, it has to cope with the phenomenon of *visual aliasing*, which means that the visual information captured from very different positions may be very similar. Taking these facts into account, to build a functional visual model of the environment and to estimate the pose (position and orientation) of the robot within this model with robustness, it is necessary to find an alternative codification which is more efficient and robust against such phenomena.

Two main frameworks can be found in the literature to extract this information based either on local or on global appearance. The first family of methods consists in detecting some outstanding landmarks or regions and describing them using any algorithm that provides some invariance against transformations, such as SIFT [10], SURF [11], BRIEF [12], BRISK [13], ORB [14], FREAK [15] and LDB [16]. The second family consists of working with each scene as a whole, trying to build a unique descriptor per image that collects information on its global structure, using some approaches such as Principal Components Analysis [17], discrete Fourier transform [18], banks of Gabor filters [19], color histograms [20,21], directly subsampled versions of the original image [22] or Radon transform [23].

Traditionally, researchers have focused on the use of local appearance methods, and it can be considered a mature technology to solve the mapping and localization problems. Many approaches are proposed in the literature based on these descriptors [24–28]. Typically, they require the implementation of detection, description and tracking algorithms which tend to be relatively complex and computationally expensive. While they are often designed to be invariant against some movements of the robot, their behavior can deteriorate when other usual phenomena are present, such as changes in lighting conditions, occlusions, noise or visual aliasing. Some comparative analyses of this kind of descriptor can be found in [29,30]. Thanks to these comparatives, an optimal description method can be chosen and tuned depending on the environment and application.

Global-appearance approaches have been applied to these areas more scarcely. Since each image is described through a unique descriptor, they usually lead to models of the environment that can be handled intuitively by a human operator. The localization process is more straightforward, based on the pairwise comparison between descriptors. Some authors have made use of such approaches in the field of mobile robots, such as [31–36]. These techniques may be useful in unstructured environments where it is difficult to extract robust landmarks. As a drawback, they have been used typically to build topological models [37,38], since no metric information can be extracted from pure global appearance (unless additional sensory information is added).

In [39], a comparative evaluation of the performance of global-appearance methods in mapping tasks was carried out. However, we have not found any work in the literature that makes a deep and systematic study of the role of global appearance in localization tasks. Therefore, the objective of this paper is two-fold. On the one hand, we have chosen six widespread and accepted families of visual description methods, and we have adapted them to be used efficiently with omnidirectional visual information, in such a way that the resulting descriptors contain useful information to retrieve relative distance and orientation efficiently. To this aim, some algorithms have been implemented to estimate the relative

position and orientation from these descriptors using purely visual information. On the other hand, we carry out a comparative evaluation of these descriptors in localization tasks and study their behavior against changes in the robot pose and other visual changes in the environment. Their relative performance has been tested and the influence of the most relevant parameters is assessed, completing the work presented in [39].

The remainder of the paper is structured as follows. Section 2 presents a state-of-the-art of global appearance description approaches and outlines the implementation of the three methods included in the evaluation. After that, in Section 3 the framework used to estimate the position and the orientation of the robot is detailed. Then, Section 4 presents the experimental setup and the set of images used in the experiments. The paper finishes with the results of the experiments, discussed in Section 5, and the conclusions and future lines of research in Section 6.

2. Global Appearance Descriptors

The objective of this section is two-fold. On the one hand, a state-of-the-art of global appearance descriptor is developed. On the other hand, a brief mathematical description of the methods included in the comparative analysis is made. Six families of global appearance methods have been chosen to be analyzed: methods based on the discrete Fourier transform (Section 2.1), on gradient orientation (Section 2.2), on the use of banks of Gabor filters (Section 2.3), on Speeded-Up Robust Features (SURF) description method (Section 2.4), on Binary Robust Independent Elementary Features (BRIEF) (Section 2.5) and on Radon transform (Section 2.6). A complete description of the methods can be found in [39–41]. However, for the sake of clarity, we have included an outline in this section.

We consider the movement of the robot is contained in the ground plane, and it captures images using an omnidirectional vision system mounted on its top. This system consists of a camera pointing towards a hyperbolic mirror, with their axes aligned and in vertical position. The complete experimental setup is presented in Section 4.

2.1. Descriptors Based on the Discrete Fourier Transform

The discrete Fourier transform (DFT) has been used by many researchers to extract the most relevant information from scenes. For example, Oliva and Torralba [19] propose using a windowed 2D Fourier transform, that permits defining some circular windows to select spatial information around some specific pixels in the scene. Ishiguro and Tsuji [42] propose an alternative approach, named Fourier Signature (FS), which is designed to be used on panoramic images. Menegatti et al. showed the robustness of this representation to build a model of an environment and to estimate the position of a vehicle using a Monte Carlo approach [18,31], in a relatively small environments and controlled conditions. Stürzl et al. [43] propose a visual homing algorithm based on the Fourier Signature, but the panoramic scene is previously reduced to a unidimensional array. Horst and Möller use it in visual place recognition [44].

The Fourier Signature (FS) permits obtaining a descriptor which is invariant against rotations of the robot in the ground plane when using panoramic images. For this reason, this is the DFT-based representation we have chosen in this comparative evaluation. The description process starts from a panoramic scene $f(x,y) \in \mathbb{R}^{N_1 \times N_2}$. Initially, the image can be subsampled to obtain a lower number of rows $k_1 < N_1$ ($k_1 = 1$ in [43]). The FS of the resulting scene $f(x,y) \in \mathbb{R}^{k_1 \times N_2}$ is the matrix $\mathbf{F}(u,y) \in \mathbb{C}^{k_1 \times N_2}$ obtained after calculating the unidimensional DFT of each row of the image. In the frequency domain, the main information is concentrated in the low frequency components, and the high frequency components tend to be more contaminated by the possible presence of noise in the original image. Taking this fact into account, by retaining the k_2 first columns and discarding the remainder, a compression effect is achieved. The new complex matrix, with k_1 rows and k_2 columns, can be expressed as a magnitudes matrix $\mathbf{A}(u,y) = \|\mathbf{F}(u,v)\|$ and an arguments matrix $\Phi(u,y)$.

Based on the shift Theorem of the unidimensional DFT, when two panoramic images have been captured from the same point on the floor, but having the robot different orientations around the vertical axis, both images present the same magnitudes matrix, and the arguments matrices can be used to estimate the relative orientation of the robot. Thanks to this property, the matrix $\mathbf{A}(u,y) = \|\mathbf{F}(u,y)\|$ can be considered as a visual descriptor of the robot position (as it is rotationally invariant), the matrix $\Phi(u,y)$ can be considered as a descriptor of the robot orientation (as it permits estimating this orientation), and the estimation of the position and the orientation can be addressed independently and sequentially.

To sum up, the position descriptor is the matrix $\mathbf{A}(u,y) \in \mathbb{R}^{k_1 \times k_2}$ and the orientation descriptor is the matrix $\Phi(u,y) \in \mathbb{R}^{k_3 \times k_4}$. In the experiments, different sizes will be considered, to test separately the influence these parameters have on the accuracy and computational cost of the localization process.

2.2. Descriptors Based on Histograms of Oriented Gradients

The Histograms of Oriented Gradients (HOG) are local descriptors that have been used typically in computer vision and image processing to solve object detection tasks. HOG was initially described by Dalal and Triggs [45], who used it to detect persons in sequences of images. Afterwards, some researchers presented an improved version both in detection and computational cost [46]. Hofmeister et al. [47] made use of HOG to solve the localization of small mobile robots from low resolution images, in visually simple environments and when the orientation of the robot is similar to the orientation it had when the corresponding map image was captured. In [48], the same authors present a comparative of HOG with other appearance descriptors, applied to the localization of small robots in reduced environments, with similar results. Aslan et al. study the ability of HOG to handle occlusion in human tracking [49]. In addition, Neumann et al. use HOG, among other descriptors, for image-based vehicle detection and localization in an autonomous car [50].

Originally, HOG is built to describe local areas of a scene. We redefine it as a global appearance descriptor, using an exhaustive set of cells that covers the whole image and permits describing the global appearance. The version of HOG included in the comparative evaluation is presented in [51], where a global version of HOG is used to carry out map building and Monte Carlo localization in a large environment. When used to describe panoramic scenes, it presents rotational invariance and it also permits estimating the orientation of the robot.

In brief, from the initial panoramic image, a position and an orientation descriptor are obtained using the HOG philosophy. From the initial panoramic image $f(x,y) \in \mathbb{R}^{N_1 \times N_2}$ the magnitude and the orientation of the gradient are obtained and stored in the matrices $\mathbf{M}(x,y)$ and $\Theta(x,y)$, respectively. From now on, some sets of cells are defined upon the matrix $\Theta(x,y)$ to build the two descriptors. On the one hand, to build the position descriptor, a set of k_5 horizontal cells, whose width is equal to N_2 pixels, without overlapping, and covering the whole image are defined. For each cell, an orientation histogram with b_1 bins is compiled. During this process, each pixel in $\Theta(x,y)$ is weighted with the magnitude of the corresponding pixel in $\mathbf{M}(x,y)$. At the end of the process, the set of histograms are appended to compose the position descriptor $\vec{h}_1 \in \mathbb{R}^{k_5 \cdot b_1 \times 1}$. On the other hand, the orientation descriptor is built using the same steps, but considering a set of overlapped vertical cells, with a height equal to N_1 pixels, width equal to l_1 and distance between two consecutive cells equal to d_1. The number of vertical cells is $k_6 = N_2/d_1$. After compiling a gradient orientation histogram for each cell, with b_2 bins and appending them, the result is the orientation descriptor $\vec{h}_2 \in \mathbb{R}^{k_6 \cdot b_2 \times 1}$.

The descriptor $\vec{h}_1$ is invariant against rotations of the robot in the ground plane so it can be considered as a visual descriptor of the robot position, and the information contained in $\vec{h}_2$ permits estimating the orientation of the robot with respect to a reference image.

2.3. Descriptors Based on Gist

The descriptors based on *gist* try to imitate the ability of the human perception system to recognize immediately a scene through the identification of specific regions stand out with respect to their neighborhood. This concept was introduced by Oliva and Torralba [52,53] with the idea of creating a low dimensional global image descriptor. More recent works make use of the concept of *prominence* together with *gist*. Siagian et al. [54] try to establish synergies between both concepts in a unique descriptor whose computational cost is relatively reduced. While these descriptors have been used thoroughly in classification tasks, the experience in mobile robotics localization is more sparse. Some related applications can be found in [55], where a localization and navigation system based on the *gist* and *prominence* concepts is presented; in [56], where *gist* descriptors, calculated over specific portions of a set of panoramic images, are used to solve a localization problem in urban areas; and in [57], where descriptors based on *gist* and dimensionally reduced by means of Principal Components Analysis are used to solve the loop closure problem in Simultaneous Localization and Mapping. In addition, Su et al. use *gist* in a localization framework to match keyframes, in combination with local descriptors to improve localization accuracy [58].

The description method we have included in this comparative analysis is based on the works of Siagian et al. [54] and is deeply described in [51]. It is built from orientation information, obtained by means of a bank of Gabor filters with different orientation, in some levels of resolution. First, two versions of the original panoramic image are considered: the original one and a new lower resolution version after applying a Gaussian low-pass filter and subsampling to a new size $0.5 \cdot N_1 \times 0.5 \cdot N_2$. After that, both images are filtered with a bank of m_1 Gabor filters whose orientations are evenly distributed between 0 and 180 deg. Finally, to reduce the amount of information, the pixels in each resulting image are grouped into blocks, by calculating the average intensity of all the pixels contained in a block. The block division is chosen in an identical fashion than in the case of HOG. First, a set of k_7 horizontal blocks is defined to obtain the position descriptor $\vec{g}_1 \in \mathbb{R}^{2 \cdot k_7 \cdot m_1 \times 1}$, which is invariant against rotations of the robot in the ground plane. Second, a set of k_8 vertical blocks with overlapping is defined to obtain the orientation descriptor $\vec{g}_2 \in \mathbb{R}^{2 \cdot k_8 \cdot m_2 \times 1}$.

2.4. Descriptors Based on Wi-SURF

SURF [11] has been considered one of the most important local descriptors and it has been used in countless works as in [59] or [32] where it is used to solve localization indoors. The present study is focused on the performance of global appearance descriptors. For this reason, we propose an adaptation which is based on the work [60], which extracts a unique, global appearance descriptor per image, using the SURF philosophy. Throughout the paper, we will refer to this descriptor as Whole Image SURF (Wi-SURF).

Wi-SURF has been used in previous works for topometric localization [61] or for place recognition [40]. These works propose to obtain a unique vector $d \in \mathbb{R}^{64}$ that contains gradient information of the entire image. Therefore, such a descriptor can be useful for place recognition, but does not contain enough information to estimate relative orientation. For this reason, we propose dividing the panoramic image into a set of evenly distributed square windows, with some overlapping between them. In each window, a SURF descriptor $d \in \mathbb{R}^{64}$ is calculated and all the descriptors are concatenated, which leads to a global-appearance descriptor. This approach will enable us to solve not only the localization but also to estimate the relative orientation of the robot, as detailed in Section 3.4. The square windows are evenly distributed following the next parameters: k_9 is the number of horizontal cells in which the panoramic image is split and sp_1 the horizontal space between consecutive windows. The number of windows per cell will depend on the images' width (512 columns in our experiments) so a total of $w_1 = \frac{512}{sp_1}$ windows per cell are calculated. The width of the square window is equal to the height of the horizontal cell. After all, the size of the descriptor is $\vec{ws} \in \mathbb{R}^{k_9 \cdot w_1 \cdot 64 \times 1}$. This final descriptor will be used to estimate both position and orientation.

2.5. Descriptors Based on BRIEF-Gist

BRIEF-gist is a global appearance descriptor based on the local descriptor Binary Robust Independent Elementary Features (BRIEF). BRIEF was presented in [12] and used for different mobile robot applications [62,63]. Based on this local descriptor, a global appearance descriptor is presented in [64]. This approach is known as BRIEF-gist and it has been used for place recognition and loop closure detection in [40]. In the present work, we adapt this descriptor to be used with panoramic images in such a way that it permits calculating both relative distance and orientation in a localization task.

To implement the BRIEF-gist descriptor, the image is divided into $k_{10} \times w_2$ windows equally sized. Then, using the BRIEF description methodology, a set of ordered pairs of pixels is defined in each window, and the intensity of the second pixel of each pair is compared to the first one. If the difference is positive a 1 is added to the global descriptor, and a 0 if the difference is negative. As a result, a boolean vector is obtained. After this process, the resulting BRIEF-gist descriptor is $\vec{bg} \in \mathbb{R}^{k_{10} \cdot w_2 \times 1}$. This final descriptor is used to estimate both position and orientation.

2.6. Descriptors Based on Radon Transform

The Radon transform was proposed in [65]. Initially, it was used in different computer vision applications as a geometric shape descriptor, as in [66,67]. More recently, the Radon transform (RT) has been adapted to describe globally omnidirectional images and its performance was tested in [41], where descriptors based on the RT were used to solve the image retrieval problem, and in [23], where these descriptors were used to estimate relative altitude from images. The main advantage of this descriptor is that it can be calculated with raw omnidirectional images, as captured by the vision system (with no panoramic transformation).

Mathematically, the Radon transform consists of describing a function in terms of the projections of its linear integrals.

After applying the Radon transform, the image is transformed into a function $r_{im}(\Phi, d)$, which is obtained after integrating the original function through several groups of parallel lines with distance to the origin d and different orientation Φ. The size of the new descriptor is $r_{im} \in \mathbb{R}^{M_x \times M_y}$, M_x is the number of orientations where $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_{M_x}\}$ and M_y is the number of parallel lines.

When the Radon transform is applied to omnidirectional images, it is specially interesting its symmetry and the fact that the descriptor is horizontally shifted when the robot rotates [68], which allows us to obtain global appearance descriptors that can be used to estimate position and relative orientation. This property can be seen in Figure 1, where four omnidirectional images are shown; three of them have been taken from the same position but with different orientation and the other one has been taken from a different position. The figure clearly shows the effect of the orientation in the Radon transform and how different the result is if the image is from another room. If the robot rotates ($\Delta\theta$) degrees, the new descriptor presents the same information as the original one, but it has been shifted s columns, $s = (\Delta\theta) \cdot (\text{M\_x})/360$. Thanks to this property, descriptors based on Radon transform contain position and orientation information of the robot.

To sum up, after applying the Radon transform to an omnidirectional image with size $N_x \times N_x$, a matrix $r \in \mathbb{R}^{\frac{360}{p_1} \times 0.5 \cdot N_x}$ is obtained. p_1 is the angle (deg.) between consecutive sets of lines along which the linear integrals are calculated. In the experiments, these matrices can be used in different ways in order to obtain proper uni-dimensional descriptors. Two different methods and different sizes will be considered to test the robustness of the descriptor in pose estimation. These methods and parameters are described in Section 3.

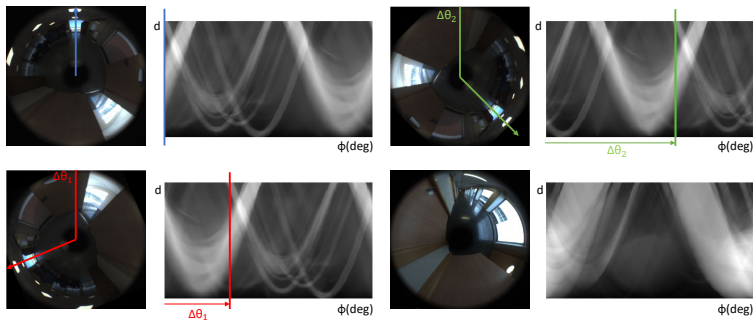


Figure 1. Shift property in the Radon transform.

3. Solving the Absolute Localization Problem

In this work, we assume a visual model of the environment is previously available. To build this model, the robot has gone through the initially unknown environment (either in a tele-operated way or using any exploration algorithm [69,70]) and has captured a set of omnidirectional images from n points of view, defined by the poses $\vec{p}_j = (x_j, y_j, \theta_j)$, $j = 1, \ldots, n$, to cover the whole environment to map. The model $\mathcal{M}$ is composed of the visual descriptors and the pose of the robot, stored for each capture position: $\mathcal{M} = \{(\mathcal{D}_1, \vec{p}_1), (\mathcal{D}_2, \vec{p}_2), \ldots (\mathcal{D}_n, \vec{p}_n)\}$ where, in general, the description of each image consists of a position and an orientation descriptor $\mathcal{D}_j = \{\vec{d}_{1j}, \vec{d}_{2j}\}$ (in the case of *Wi-SURF* and *BRIEF-gist* the same vector is used as position and orientation descriptor, so $\mathcal{D}_j = \{\vec{d}_{1j}\}$). The map building process using global appearance methods and omnidirectional imaging is thoroughly described in [39].

Once the model is built, the localization problem consists of estimating the pose of the robot. The problem is approached here as an absolute localization problem, i.e., no information on the previous position of the robot is considered, and only visual information is used. The robot captures a new image at time instant t, from an unknown pose (f_t, *test image*). Then, the descriptor of this image $\mathcal{D}_t$ is computed and compared with the set of descriptors stored in the model. From this comparison, the position and orientation of the robot at time instant t are estimated. The next subsections detail these processes depending on the description method used.

3.1. Descriptors Based on the Discrete Fourier Transform

When a test image arrives, $\mathbf{A}_t$ and Φ_t are calculated. Since the position descriptor is invariant against rotations of the robot in the ground plane, first, $\mathbf{A}_t$ is used to estimate the position of the robot, by comparing it with the descriptors $\mathbf{A}_j$, $j = 1, \ldots, n$ and retaining the k-nearest neighbors. The position of the nearest neighbor (x_i, y_i) (i is the index of the nearest neighbor) can be considered as an estimation of the position of the robot at time instant t. Once the position of the robot has been estimated, the arguments matrix of the *test image*, Φ_t, and the arguments matrix of the nearest neighbor, Φ_i, are used to estimate the orientation of the robot, using the shift theorem of the DFT. The objective is to estimate the relative orientation θ_{ti} of the robot at time instant t with respect to the orientation the robot had when capturing the nearest neighbor, $\theta_{ti} = \theta_t - \theta_i$. The next steps are as follows:

1. A set of artificial rotations is applied to the *test image*. The shift theorem of the unidimensional DFT can be used to generate the argument matrices of the test image rotated siblings. The step between consecutive rotations is $\Delta\phi$. This is equivalent to making a shift of the columns of the panoramic image with a magnitude of d pixels, where $\Delta\phi = d \cdot 2\pi / N_2$. In the experiments, we consider $d = \{1, 2, \ldots, N_2 - 1\}$. This means that the angular step between consecutive artificial rotations is $\Delta\phi = 2\pi / N_2$. This is the resolution of the method.

2. After this process, a set of $n_{rot} = 2\pi/\Delta\phi$ arguments matrices are available at time instant t.

$$\{\Phi_0, \Phi_1, \ldots, \Phi_{n_{rot}}\}_t = \{\Phi_\alpha\}_t, \alpha = 0, \ldots, n_{rot} \quad (1)$$

3. The Hadamard product of the matrix Φ_t and every matrix Φ_α is calculated. The sum of the components of each resulting matrix is obtained, and the result is an array of data:

$$\{m_0, m_1, \ldots, m_{n_{rot}}\}_t = \{m_\alpha\}_t, \alpha = 0, \ldots, n_{rot} \quad (2)$$

4. The estimated relative rotation is the α value whose coefficient m_α presents the maximum value.

$$\alpha = arg\,max_\alpha\{m_\alpha\} \quad (3)$$

$$\theta_{ti} = \frac{2\pi\alpha}{n_{rot}} \quad (4)$$

where θ_{ti} is the relative orientation between the image im_t and the nearest neighbor of the map, im_i. This way, the absolute orientation of the robot at time instant t can be calculated as:

$$\theta_t = \theta_i + \theta_{ti} \quad (5)$$

In this equation, θ_i is the orientation that the robot had when the map image im_i was captured, with respect to the global reference system.

In the experiments, the parameters of the Fourier Signature to optimize are the size of the module matrix (k_1 and k_2) and the size of the arguments matrix (k_3 and k_4) to reach a balance between the accuracy in the estimation of the position and orientation and the computational cost of the algorithms.

3.2. Descriptors Based on Histograms of Oriented Gradients

Once the test image im_t has been captured, the descriptors $\vec{h}_{1t}$ and $\vec{h}_{2t}$ are calculated. First, the k-nearest neighbors to $\vec{h}_{1t}$ among the set of descriptors $\vec{h}_{1j}$, $j = 1, \ldots, n$ are calculated and extracted. The position (x_i, y_i) of the nearest neighbor i is an estimation of the position of the robot at time instant t.

Later, the orientation is calculated by comparing the vector $\vec{h}_{2t}$ with the vector $\vec{h}_{2i}$. With this aim, a set of artificial rotations is calculated using the vector $\vec{h}_{2t}$ and later, the scalar product between the resulting vector after each rotation and the vector $\vec{h}_{2i}$ is calculated. To simulate a rotation of the vector $\vec{h}_{2t}$, the circular shift must be a multiple of b_2 positions (b_2 is the number of bins per histogram). A shift of b_2 positions equals a rotation of the robot $\Delta\phi = 2\pi d_1/N_2$ radians (this is the angular resolution of the method), where d_1 is the distance between two consecutive vertical cells.

Finally, the estimated relative orientation θ_{ti} of the robot is the angle that corresponds to the rotated version of the vector $\vec{h}_{2t}$ which presents a higher scalar product with $\vec{h}_{2i}$.

3.3. Descriptors Based on Gist

The processes to estimate the position and orientation are identical to those presented in the case of HOG. Once captured the test image im_t, the descriptors $\vec{g}_{1t}$ and $\vec{g}_{2t}$ are calculated. First, $\vec{g}_{1t}$ is compared to $\vec{g}_{1j}$, $j = 1, \ldots, n$ and the k-nearest neighbors are calculated. From them, the position $(x, y)_i$ of the nearest neighbor i is considered an estimation of the position of the robot at time instant t. After that, the orientation is calculated by comparing the vector $\vec{g}_{2t}$ with the vector $\vec{g}_{2i}$. With this aim, successive artificial rotations are calculated, using the vector $\vec{g}_{2t}$ and later, the scalar product between

each rotated version and the vector $\vec{g}_{2i}$ is obtained. To make an artificial rotation of the vector $\vec{h}_{2t}$, the magnitude of the circular shift must be a multiple of m_2 (m_2 is the number of components of each vertical block). Every shift equals a rotation of $\Delta\phi = 2\pi d_2/N_2$ radians (this is the angular resolution of the method), where d_2 is the distance between two consecutive vertical blocks in the descriptor.

The resulting orientation θ_{ti} is the angle that corresponds to the rotated version of $\vec{g}_{2t}$ that presents the highest scalar product with $\vec{g}_{2i}$.

3.4. Descriptors Based on Wi-SURF

Once the test image im_t is taken, the descriptor $\vec{ws}_t$ is obtained. First, this descriptor is compared with the descriptors $\vec{ws}_j, j = 1, \ldots, n$, to calculate the relative orientation between the test descriptor and the descriptors in the model. To estimate the relative orientation, some artificial rotations are added to $\vec{ws}_t$ and the distance between the resulting descriptor after each rotation and $\vec{ws}_j$ is calculated. To simulate an artificial rotation of $\vec{ws}_t$, a circular shift is applied, which must be a multiple of 64 positions (the SURF descriptor of each window contains 64 components) and w_1 (number of windows). The 64-position shift of the descriptor equals to a rotation of the robot $\Delta\phi = 2 \cdot \pi \cdot sp_1/N_2$ radians (and therefore, this is the angular resolution of the method). Once the relative orientation between the test descriptor and each of the descriptors in the model has been calculated, each descriptor $\vec{ws}_j$ is shifted in such a way that the resulting descriptor has the same orientation as $\vec{ws}_t$.

Once all the descriptors are supposed to be in the same orientation, the k-nearest neighbors to $\vec{ws}_t$ are calculated among the set of descriptors in the model (once they are equally oriented with respect to $\vec{ws}_t$). The position (x_i, y_i) of the nearest neighbor i is an estimation of the position of the robot at time t. The orientation between them has been calculated previously and the corresponding angle θ_{ti} is the relative orientation estimated between the test vector and the vector evaluated from $\vec{ws}_j$.

3.5. Descriptors Based on BRIEF-Gist

Firstly, the relative orientation between images is estimated. The descriptor $\vec{bg}_t$ is calculated from the test image im_t, and the relative orientation between it and each of the descriptors $\vec{bg}_j, j = 1, \ldots, n$ is estimated. To estimate it, successive artificial rotations are applied to $\vec{bg}_t$, the scalar product between the resulting vector after each rotation and $\vec{bg}_j$ is calculated and the minimum is retained. To simulate an artificial rotation of $\vec{bg}_t$, the circular shift must be a multiple of w_2 (number of windows in each cell). As explained in Section 2.5, to calculate this descriptor the image is divided into $k_{12} \times w_2$ windows, so the angular resolution of the method is determined by the number of windows w_2. Every w_2 shift is equal to a rotation of the robot $\Delta\phi = 2 \cdot \pi/w_2$ radians.

After estimating the relative orientation, each descriptor in the model $\vec{bg}_j$ is rotated such that the resulting descriptor has the same orientation than $\vec{bg}_t$. Then the k-nearest neighbors to $\vec{bg}_t$ are calculated among the set of rotated descriptors in the model. The position (x_i, y_i) of the nearest neighbor i is an estimation of the position of the robot at time t. The relative orientation between them has been calculated previously and the corresponding angle θ_{ti} is the difference of orientation estimated between the test vector and the vector evaluated from $\vec{ws}_j$.

3.6. Descriptors Based on the Radon Transform

In the present work, we process this descriptor using two different methods to retrieve both position and orientation.

3.6.1. Radon–Fourier Method

After applying the Radon transform, a matrix $r \in \mathbb{R}^{\frac{360}{p_1} \times 0.5 \cdot N_x}$ is obtained. Then, the Fourier Signature of this matrix is calculated. As a result of this second transformation,

a matrix of magnitudes $\mathbf{A_{RT}}_j \in \mathbb{R}^{\frac{360}{p_1} \times k_{11}}$ and a matrix of arguments $\Phi_{RTj} \in \mathbb{R}^{\frac{360}{p_1} \times k_{12}}$ are obtained. As in the case of the descriptors based on the DFT, $\mathbf{A_{RT}}_j$ is used as position descriptor and Φ_{RTj} is used as an orientation descriptor. k_{11} is the number of columns taken for the position descriptor $\mathbf{A_{RT}}_j$ and k_{12} is the number of columns taken for the orientation descriptor Φ_{RTj}. To estimate the position and orientation, we use the same process as in the descriptors based on the discrete Fourier transform, presented in the Section 3.1.

3.6.2. Radon–POC Method

This method uses directly the matrix obtained after applying the Radon transform ($r \in \mathbb{R}^{\frac{360}{p_1} \times 0.5 \cdot N_x}$) as the image descriptor r_{poc_j}. To compare two descriptors, Phase Only Correlation (POC) is used. This operation outputs a correlation coefficient that allows us to estimate the similarity between two matrices and their relative shift.

To sum up, Table 1 shows the parameters whose influence will be studied in the comparative evaluation. After that, Table 2 gives details of the contents of the model when we consider each description method.

Table 1. Parameters whose influence in the localization process is studied.

Descriptor	Parameters
FS	$k_1 \Rightarrow$ number of rows, position descriptor $\mathbf{A}_j$ $k_2 \Rightarrow$ number of columns, position descriptor $\mathbf{A}_j$ $k_3 \Rightarrow$ number of rows, orientation descriptor Φ_j $k_4 \Rightarrow$ number of columns, orientation descriptor Φ_j
HOG	$b_1 \Rightarrow$ number of bins per histogram, position descriptor $\vec{h}_{1j}$ $k_5 \Rightarrow$ number of horizontal cells, position descriptor $\vec{h}_{1j}$ $b_2 \Rightarrow$ number of bins per histogram, orientation descriptor $\vec{h}_{2j}$ $l_1 \Rightarrow$ width of vertical cells, orientation descriptor $\vec{h}_{2j}$ $d_1 \Rightarrow$ distance between vertical cells, orientation descriptor $\vec{h}_{2j}$ $k_6 = \frac{N_2}{d_1} \Rightarrow$ number of vertical cells, orientation descriptor $\vec{h}_{2j}$
Gist	$m_1 \Rightarrow$ number of orientations (Gabor filters), position descriptor $\vec{g}_{1j}$ $k_7 \Rightarrow$ number of horizontal blocks, position descriptor $\vec{g}_{1j}$ $m_2 \Rightarrow$ number of orientations (Gabor filters), orientation descriptor $\vec{g}_{2j}$ $l_2 \Rightarrow$ width of vertical blocks, orientation descriptor $\vec{g}_{2j}$ $d_2 \Rightarrow$ distance between vertical blocks, orientation descriptor $\vec{g}_2$ $k_8 = \frac{N_2}{d_2} \Rightarrow$ number of vertical blocks, orientation descriptor $\vec{g}_{2j}$
WS	$w_1 \Rightarrow$ number of windows per cell, descriptor $\vec{ws}_j$ $k_9 \Rightarrow$ number of horizontal blocks, descriptor $\vec{ws}_j$ $sp_1 \Rightarrow$ horizontal space between windows, descriptor $\vec{ws}_j$
BG	$w_2 \Rightarrow$ number of windows per cell, descriptor $\vec{bg}_j$ $k_{10} \Rightarrow$ number of horizontal blocks, descriptor $\vec{bg}_j$
RT	$p_1 \Rightarrow$ degrees between lines where Radon is calculated, matrix r $k_{11} \Rightarrow$ number of columns, position descriptor $\mathbf{A_{RT}}_j$ $k_{12} \Rightarrow$ number of columns, orientation descriptor Φ_{RTj} $N_x \Rightarrow$ omnidirectional images' size is $N_x \times N_x$

Table 2. Contents of the map, for localization and orientation estimation, per image included in the model im_j, $j = 1, \ldots, n$.

Descriptor	Localization	Orientation
FS	$\mathbf{A}_j \in \mathbb{R}^{k_1 \times k_2}$	$\mathbf{\Phi}_j \in \mathbb{R}^{k_3 \times k_4}$
HOG	$\vec{h}_{1j} \in \mathbb{R}^{k_5 \cdot b_1 \times 1}$	$\vec{h}_{2j} \in \mathbb{R}^{k_6 \cdot b_2 \times 1}$
Gist	$\vec{g}_{1j} \in \mathbb{R}^{2 \cdot k_7 \cdot m_1 \times 1}$	$\vec{g}_{2j} \in \mathbb{R}^{k_8 \cdot m_2 \times 1}$
WS	$\vec{ws}_j \in \mathbb{R}^{k_9 \cdot w_1 \cdot 64 \times 1}$	
BG	$\vec{bg}_j \in \mathbb{R}^{k_{10} \cdot w_2 \times 1}$	
RT–F	$\mathbf{A_{RT}}_j \in \mathbb{R}^{\frac{360}{p_1} \times k_{11}}$	$\mathbf{\Phi}_{RTj} \in \mathbb{R}^{\frac{360}{p_1} \times k_{12}}$
RT–POC	$r \in \mathbb{R}^{\frac{360}{p_1} \times 0.5 \cdot N_x}$	

4. Experimental Setup

This section describes the experimental setup. First, the sets of images used to carry out the experiments are presented. Second, a variety of phenomena (noise and occlusions) to test the robustness of the algorithms are described.

4.1. Sets of Images

All the experiments are carried out with two sets of images captured by ourselves [71]. A catadioptric vision system is used to capture the images. It is composed of an *Imaging Source DFK 21BF04* camera pointing towards an *Eizoh Wide 70* hyperbolic mirror, with their axes aligned. This system captures omnidirectional images which are preprocessed to obtain cylindrical projections (panoramic images) with size $N_1 \times N_2 = 128 \times 512$ pixels.

The first set of images is named the *training set* and it is composed of 872 panoramic images captured on a dense grid of points of 40×40 cm, covering the whole floor of a building of Miguel Hernández University (Spain), including 6 different rooms. The *training set* will be used to build a visual model of the environment. Different grid sizes will be considered along the experiments.

The second set is named the *test set* and it contains 1232 images captured in all the rooms, with different orientations. To capture these images, 77 positions were defined on some half-way points among the grid positions, and 16 images per position were captured, with different robot orientations, to cover the whole circumference. These images were captured in different times of day and with changes in the position of some objects, doors, etc., to reflect the natural variability of the visual information in real working environments. The *test set* will be used during the process of localization and orientation estimation, to test the goodness of each description method and the influence of the main parameters. This environment is very prone to *perceptual aliasing*, which means that two images captured from two positions which are far away may have a similar visual appearance. Global appearance descriptors must cope with this phenomenon as it frequently happens in indoor environments.

Figure 2 shows a bird's eye view of the environment and the capture points of the training images. As an example, Figure 3 shows the library, the capture points of the training (red) and test (green) images and some sample training and test images captured in close points. The effect of changes in lighting conditions and changes of orientation can be appreciated. Other sample space is shown in Figure 4 (corridor). The effect of *visual aliasing* is clearly shown. In addition, the test image 3 shows an example of changes in the environment (open door with respect to the training images).

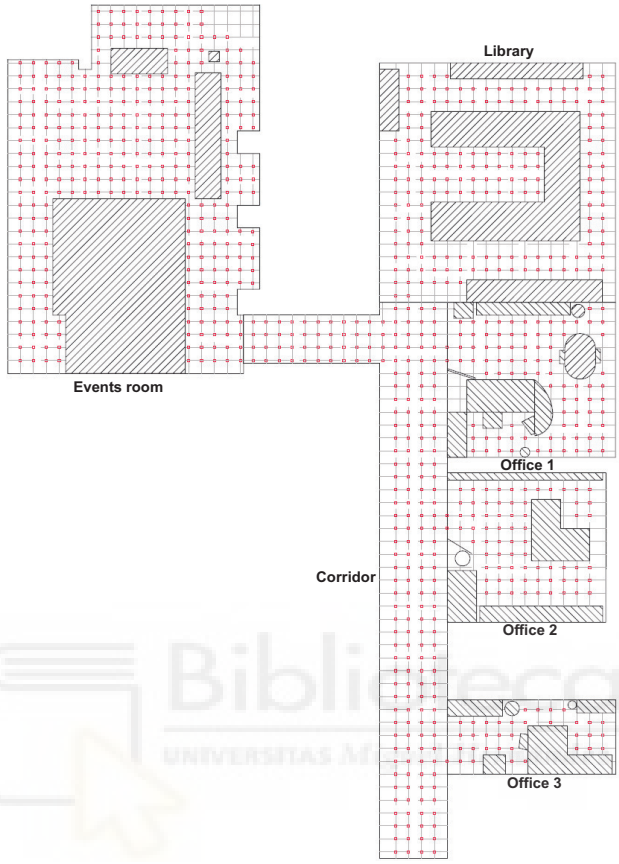


Figure 2. Bird's eye view of the capture points of the training set of images. The size of the grid is 40 × 40 cm.

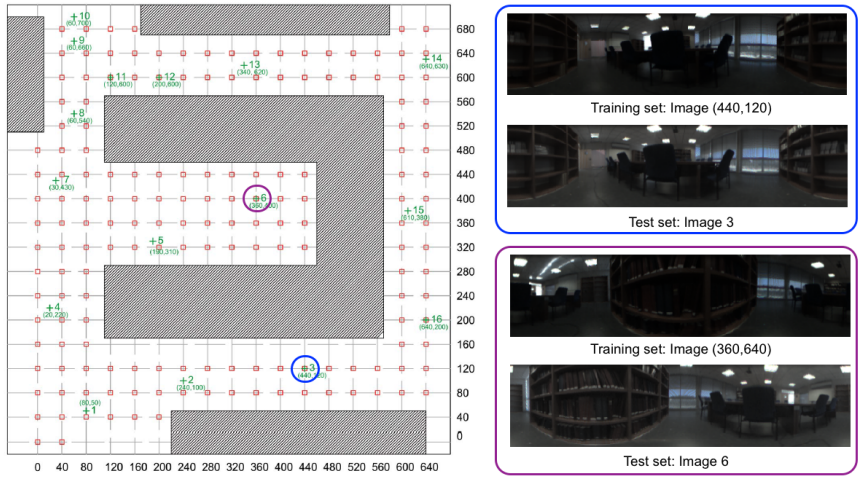


Figure 3. Library. Bird's eye view of the capture points of the training set of images. The size of the grid is 40 × 40 cm.

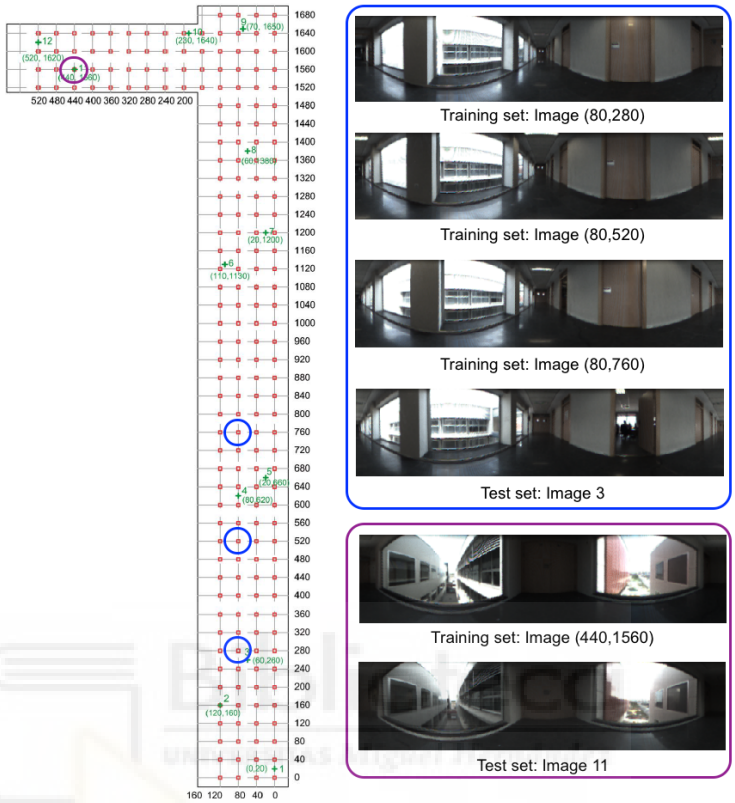


Figure 4. Corridor. Bird's eye view of the capture points of the training set of images. The size of the grid is 40 × 40 cm.

4.2. Addition of Noise and Occlusions

The test images reflect some of the most habitual undesired effects in real working environment: changes in lighting conditions, in the position and state of some objects and perceptual aliasing. Additionally, two other phenomena are considered in the experiments: noise and occlusions.

First, to test the influence of noise due to the nature of the acquisition system, noise with Gaussian distribution is considered, with null average value and several variance values, to consider different noise levels: $\sigma^2 = \{0, 0.0025, 0.05, 0.01, 0.02, 0.05\}$. Along the rest of the paper, these levels of noise are named noises 0, 1, 2, 3, 4 and 5, respectively. Figure 5a shows a test image with these levels of added noise. In the most extreme case, the visual appearance of the image is seriously altered.

Second, the presence of persons or other robots in the environment may occlude partially and temporarily the visual information. Working with panoramic images constitutes an advantage as far as occlusions are concerned. However, they may hide some relevant features with respect to the visual information stored in the map and put in risk the localization process. To model this effect, several levels of occlusion have been added artificially to the images, considering several vertical bars that produce different levels of occlusion, considering $\{0, 5, 10, 20, 40\}$% of the whole image occluded. Along the rest of the paper, these levels are named occlusions 0, 1, 2, 3, 4 and 5, respectively. Figure 5b shows a test image with these levels of added occlusion. In the most extreme case, 40% of the visual information is lost.

(a) (b)

Figure 5. Sample image from the training test with (**a**) different levels of added Gaussian noise ($\sigma^2 = \{0, 0.0025, 0.05, 0.01, 0.02, 0.05\}$) and (**b**) sequence of occlusions considered ($\{0, 5, 10, 20, 40\}$%).

5. Results and Discussion

In this section, an exhaustive bank of experiments is proposed to test the performance of the global appearance descriptors included in the comparative evaluation and the influence of the main parameters in the accuracy and computational cost of the localization process. The experiments have been structured in four subsections. First, in Section 5.1, the ability of each descriptor to find the nearest neighbor of the model, in ideal conditions (considering neither noise nor occlusions) is tested. After that, the problem of position estimation is solved, including also the study of performance with these effects (Section 5.2). Third, in Section 5.3, the problem of orientation estimation is considered. Finally, Section 5.4 studies the relative performance of the descriptors with a trajectory-like dataset.

5.1. Image Retrieval Problem

During the localization process, the first step consists of comparing the localization descriptor of the test image with all the localization descriptors in the map and obtaining the k-nearest neighbors. Taking this fact into account, in this section we evaluate the ability of each description method to calculate correctly the first nearest neighbor (i.e., to identify correctly the position of the model which is geometrically the nearest one to the test position). It is known as the *image retrieval problem*.

To obtain the k-nearest neighbors of a test image descriptor, several kinds of distances can be considered. In this study, four distance measurements are implemented and compared. In the next lines, these distances are formalized. Considering $\vec{r} = \{r_i\}$, $i = 1, \ldots, l$ and $\vec{s} = \{s_i\}, i = 1, \ldots, l$, the two data vectors whose distance we want to obtain:

1. Weighted metric distance:

$$dist_p(\vec{r}, \vec{s}) = \left(\sum_{i=1}^{l} \omega_i \cdot |r_i - s_i|^p \right)^{\frac{1}{p}} \quad (6)$$

If we consider $\omega_i = 1$, $i = 1, \ldots, l$, the Minkowski distance is obtained. Two particular cases will be considered: $dist_1$ (*Manhattan* distance), which is defined from the Minkowski distance with $p = 1$, and $dist_2$ (Euclidean distance), doing $p = 2$.

2. Pearson correlation coefficient. It is a similitude coefficient that can be obtained as:

$$sim_{Pea}(\vec{r}, \vec{s}) = \frac{\vec{r}_d^T \cdot \vec{s}_d}{|\vec{r}_d||\vec{s}_d|} \tag{7}$$

where $\vec{r}_d = [r_1 - \bar{r}, \ldots, r_l - \bar{r}]$ and $\vec{s}_d = [s_1 - \bar{s}, \ldots, s_l - \bar{s}]$, $\bar{r} = \frac{1}{l}\sum_j r_j$, $\bar{s} = \frac{1}{l}\sum_j s_j$. It takes values in the range $[-1, +1]$. From this similitude coefficient, a distance measure can be defined as:

$$dist_3(\vec{r}, \vec{s}) = 1 - sim_{Pea}(\vec{r}, \vec{s}) \tag{8}$$

3. Inner product: It is also a similitude coefficient that can be calculated as the scalar product between the two vectors to compare.

$$sim_{cos}(\vec{r}, \vec{s}) = \frac{\vec{r}^T \cdot \vec{s}}{|\vec{r}||\vec{s}|} \tag{9}$$

As shown in the equation, $\vec{r}$ and $\vec{s}$ are usually normalized. In this case, this measure is known as *cosine similitude* and takes values in the range $[-1, +1]$. The corresponding distance value is:

$$dist_4(\vec{r}, \vec{s}) = 1 - sim_{in}(\vec{r}, \vec{s}) \tag{10}$$

Therefore, the four distance measurements compared along this section are: $dist_1$ (Manhattan distance), $dist_2$ (Euclidean distance), $dist_3$ (Pearson correlation-based distance) and $dist_4$ (cosine similitude-based distance).

First, the success rate of each algorithm is studied. It assesses the ability of the localization algorithm to calculate correctly the first nearest neighbor (i.e., to identify correctly the position of the model which geometrically the nearest one to the test position). Figures 6–12 show the success rate, expressed on a per unit base. For comparative purposes, all the results are expressed in the same color scale.

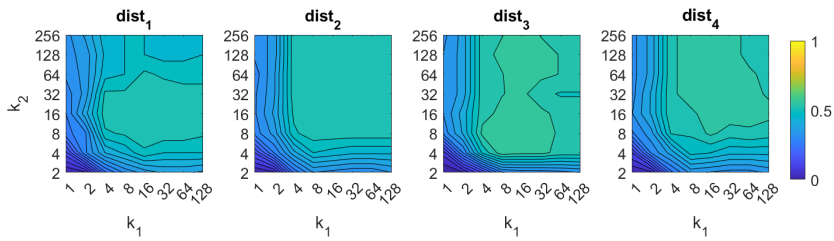


Figure 6. FS image retrieval problem. Success rate of the method. k_1 and k_2 are, respectively, the number of rows and columns of the descriptor (Table 1).

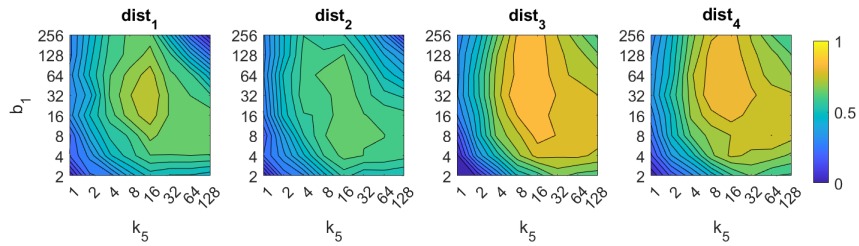


Figure 7. HOG image retrieval problem. Success rate of the method. k_5 is the number of horizontal cells and b_1 the number of bins per histogram (Table 1).

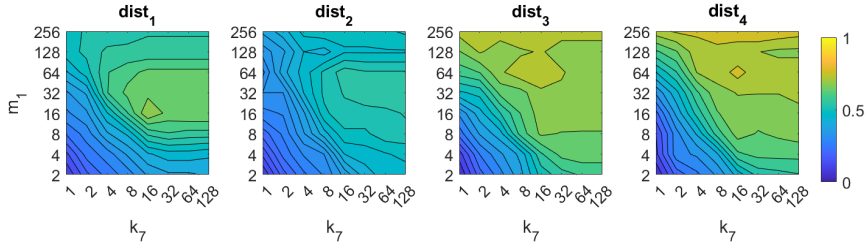


Figure 8. *Gist* image retrieval problem. Success rate of the method. k_7 is the number of horizontal blocks and m_1 the number of Gabor filters to build the descriptor (Table 1).

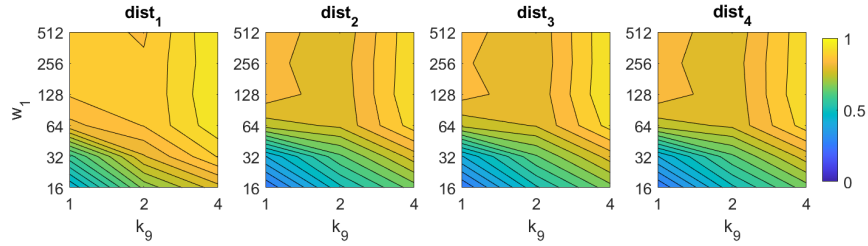


Figure 9. WS image retrieval problem. Success rate of the method. k_9 is the number of horizontal cells and w_1 the number of windows per cell (Table 1).

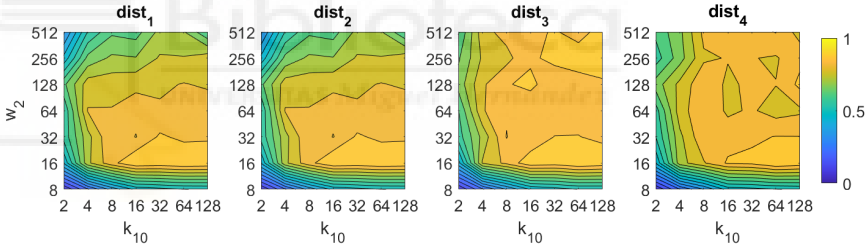


Figure 10. BG image retrieval problem. Success rate of the method. k_{10} is the number of horizontal cells and w_2 the number of windows per cell (Table 1).

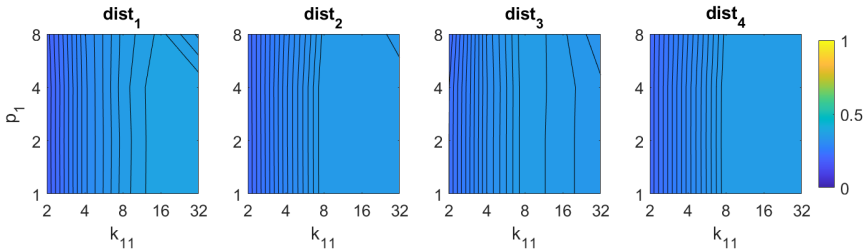


Figure 11. RT–F image retrieval problem. Success rate of the method. k_{11} is the number of blocks and p_1 the relative angle (deg) between the lines in each set (Table 1).

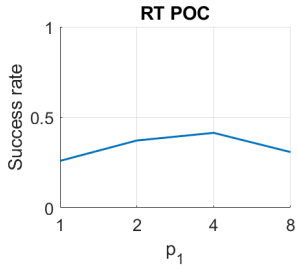


Figure 12. RT–POC image retrieval problem. Success rate of the method. p_1 is the relative angle (deg) between the lines en each set (Table 1).

The behavior of the FS changes slightly depending on the distance measurement used. The best results are obtained with $dist_3$ and $dist_4$ with an intermediate number of rows and an intermediate to high number of columns. In all cases, an excessively low number of rows and/or columns provides bad results. The best accuracy is 60%, and it is obtained with the distance $dist_3$ and $k_1 = k_2 = 8$.

About HOG, the best results are also obtained with distances $dist_3$ and $dist_4$. In both cases, the number of horizontal cells k_5 must be an intermediate value, around 16. A higher number does not improve the accuracy of the method. The number of bins per histogram b_1 must take values from intermediate to high, starting from 16. In the case of distances $dist_1$ and $dist_2$, an excessively high number of cells and bins also provides remarkably bad results. The best accuracy is 89%, and it is obtained with the distance $dist_3$ and $k_5 = 8, b_1 = 32$.

In the case of *gist*, the best results are obtained again using the distances $dist_3$ and $dist_4$. In these cases, the accuracy increases as the number of masks m_1 does. It is not necessary a high number of masks m_1 to obtain good results. The best accuracy is 89%, and it is obtained with the distance $dist_3$ and $k_7 = 32, m_1 = 256$.

In the case of *Wi-SURF*, the best results are obtained using the distances $dist_1$ and $dist_3$. In these cases, the image retrieval problem is solved with a better rate when using high values of k_9 (around 4). The process performs correctly with intermediate and high number of windows per cell w_1, starting from 128. The best rate is 97%, and it is obtained with the distance $dist_1$ and $k_9 = 4, w_1 = 512$.

If we analyze now *BRIEF-gist*, the best results are obtained using the distances $dist_3$ and $dist_4$. A high number of horizontal cells k_{10} is needed to obtain suitable results, about 64. A high number of windows w_2 does not improve the results necessarily, but remarkably bad results are obtained using low values of k_{10} or w_2. The best accuracy obtained with *BG* is 93%, and it is obtained with the distance $dist_3$ and $k_{10} = 64, w_2 = 16$.

Finally, in the case of *RT*, the results are not competitive if they are compared with the rest of the descriptors. On the one hand, using the Radon transform along with the Fourier Signature, the best results are obtained with the distances $dist_1$ and $dist_4$. In this case the parameters have less relevance on the results, but in general, high values of k_{11} and low values of p_1 lead to better rates. Using *RT–F*, the best accuracy is 39%, and it is obtained with the distance $dist_1$ and $k_{11} = 32, p_1 = 1$. On the other hand, using the POC method, the best rate is 41% obtained with $p_1 = 4$.

Analyzing globally these figures, Wi-SURF is the description algorithm that presents the best absolute success rate, when it is used along with $dist_1$. In general, the distance $dist_3$ performs much better than the rest in almost all the cases. HOG, *gist* and *BRIEF-gist* are also acceptable methods. Taking into account the challenging characteristics of the environment, they provide remarkably good results.

Apart from the success rate, it is also worth studying the computational cost of the process, to evaluate whether the localization task could be carried out in real time. Figures 13–19 show the necessary time to obtain the nearest neighbor, depending on the size of the position descriptor. The average value after all the experiments is shown,

expressed in seconds. A logarithmic scale has been used to represent efficiently the time in the color scale.

The experiments have been carried out with a CPU Intel Core i7-9700 at 3 GHz and using the mathematical tool Matlab. These time results are not absolute, they depend of the computer which runs the process. They are comparable because all the calculations have been done with the same machine.

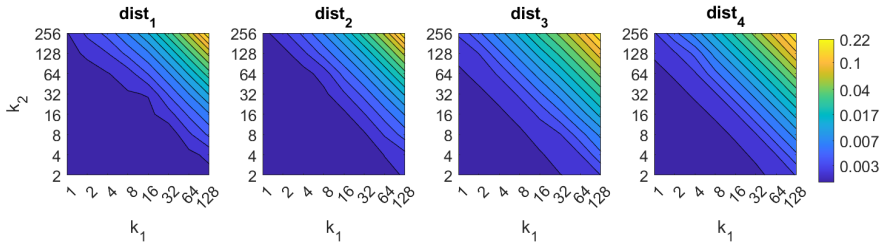


Figure 13. FS image retrieval problem. Computational time.

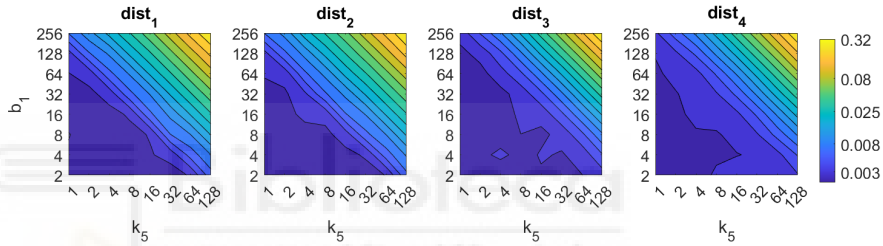


Figure 14. HOG image retrieval problem. Computational time.

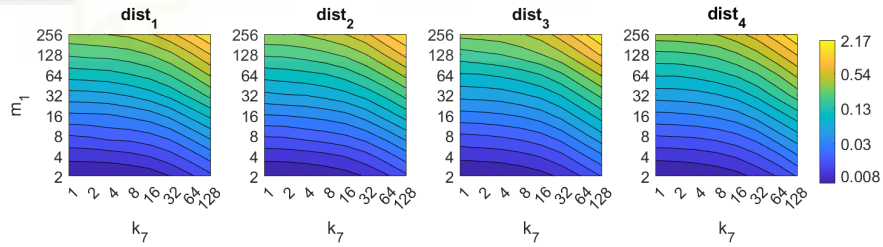


Figure 15. *Gist* image retrieval problem. Computational time.

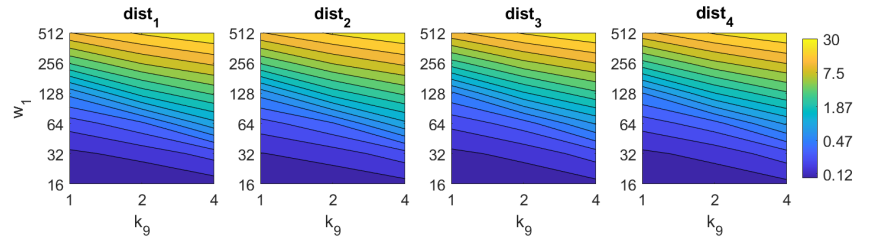


Figure 16. WS image retrieval problem. Computational time.

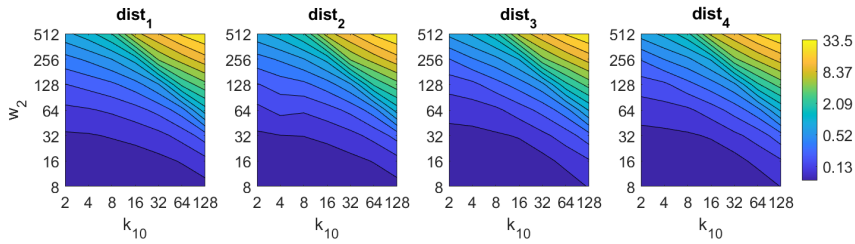


Figure 17. BF image retrieval problem. Computational time.

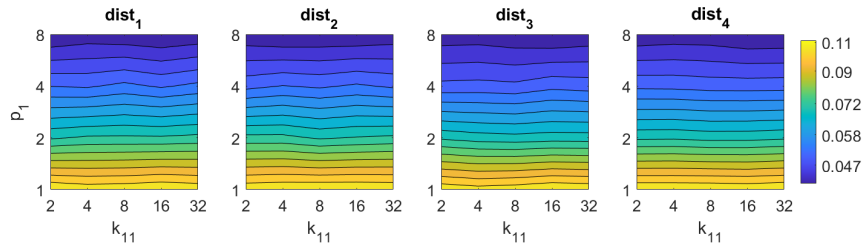


Figure 18. RT–F image retrieval problem. Computational time.

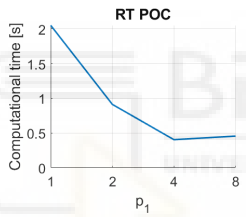


Figure 19. RT–POC image retrieval problem. Computational time.

First, FS is the quicker algorithm. The average time per *test image* is under 0.02 s for the majority of configurations. Only when both k_1 and k_2 take high values, the computational time takes values around 0.22 s. Both parameters have a similar influence on the computational cost. Second, the computational cost of HOG is slightly higher than FS, depending on the configuration of the parameters. Both parameters b_1 and k_3 have similar influence on this time. When their values are high it is possible to find some results where the runtime takes around 0.32 s. Third, *gist* is computationally more an expensive algorithm. m_1 has a strong influence on the necessary time. A high number of masks along with high values of k_7 make the time per image to take values around 2.1 s. Anyway, it is possible to find configurations that provide acceptable computational times with a lower number of components.

The second group of descriptors, in which each descriptor should be shifted until finding the relative orientation before retrieving the image, are considerably slower. On the one hand, *Wi-SURF* needs more than 2 s with most of the configurations. w_1 has more influence on the computational time so, as far as possible, it is better to avoid high values of this parameter. High values of the parameters can lead to times up to 30 s. On the other hand, *Wi-SURF* is the computationally most expensive method. w_2 has a strong influence on the process, and produces times about 33.5 s.

Finally, the method based on the Radon transform and Fourier performs quickly, with times typically under 0.1 s. The method based on Radon transform and POC leads to times around 0.5 s with some configurations of p_1. Notwithstanding that, since the descriptors based on the Radon transform have proved to perform poorly in the image retrieval task, these descriptors are not included in subsequent analyses.

5.2. Estimation of the Position

The second set of experiments assesses the ability of each description method to estimate correctly the position of the robot, when noise or occlusions are present, depending on the size of the descriptor and the type of distance considered.

For each test image, the position descriptor is obtained and compared with all the position descriptors in the map. The 1st nearest neighbor is then retained, using any distance measurement. In the cases that it is possible, a k-d tree has been implemented to make efficiently this search. After obtaining the nearest neighbor, the Euclidean distance between the real position of the robot at time instant t and the position of the nearest neighbor is considered the position error.

Figures 20 and 21 present the results obtained with the Fourier Signature considering the presence of noise or occlusions, respectively, in the test images. In these figures, first, no filter is considered (a) and second, a homomorphic filtering is carried out both to the reference and the test images (b). The result is expressed then as the average position error, expressed in cm after considering the 1232 test images. The horizontal axis expresses the percentage of information considered per configuration, expressed in logarithmic scale. The ticks of each graphical representation are $\{2^{-15}, 2^{-14}, 2^{-13}, \ldots, 2^{-2}, 2^{-1}\}$ which correspond, respectively, to the next percentages of information $\{0.003\%, 0.06\%, 0.12\%, \ldots, 25\%, 50\%\}$. These percentages express the information contained in each descriptor with respect to the information contained in each original panoramic image $\left(\frac{k_1 \cdot k_2}{N_1 \cdot N_2} \cdot 100\right)$. In general, the use of homomorphic filtering worsens the results. As expected, the higher the level of noise, the higher the error. However, $dist_1$ and $dist_2$ present a more robust behavior when noise is present. About the presence of occlusions, the FS descriptor is quite sensitive to this phenomenon and the results worsen substantially when the percentage of occlusion increases.

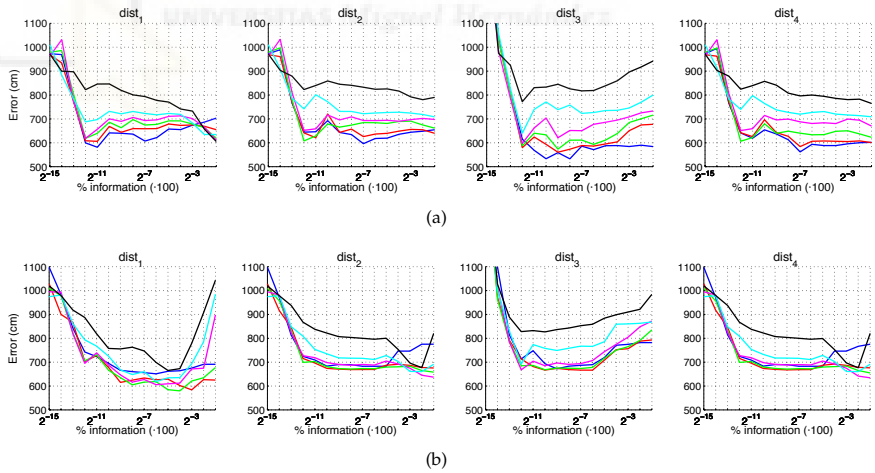


Figure 20. FS average localization error with noise: (**a**) no filter and (**b**) homomorphic filtering. Legend: — Original, — Noise 1, — Noise 2, — Noise 3, — Noise 4, — Noise 5.

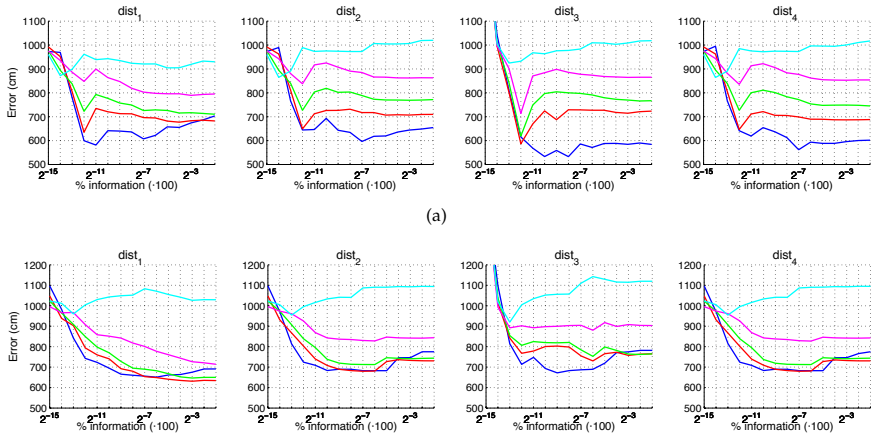


Figure 21. FS average localization error with occlusions: (**a**) no filter and (**b**) homomorphic filtering. Legend: — Original, — Occlusion 1, — Occlusion 2, — Occlusion 3, — Occlusion 4.

Figures 22 and 23 present the results obtained with the Histogram of Oriented Gradients considering the presence of noise or occlusions, respectively, in the test images. In these figures, first, no filter is considered (a) and second, a homomorphic filtering is carried out both to the reference and the test images (b). Like in the case of FS, the percentages in the horizontal axis express the information contained in each descriptor with respect to the information contained in each panoramic image. In the case of HOG they can be obtained as $\left(\frac{k_5 \cdot b_1}{N_1 \cdot N_2} \cdot 100\right)$. In presence of noise, the use of homomorphic filtering only improves the results with distances $dist_3$ and $dist_4$ and with low level of noise. Intermediate percentages of information tend to present the best absolute results so it is not necessary to store a big quantity of information during the construction of the descriptor. In presence of noise, the best absolute results are obtained with $dist_3$, no filter and intermediate quantity of information. Comparing to the other description methods, HOG stands out thank to its robustness against presence of occlusions in the test images.

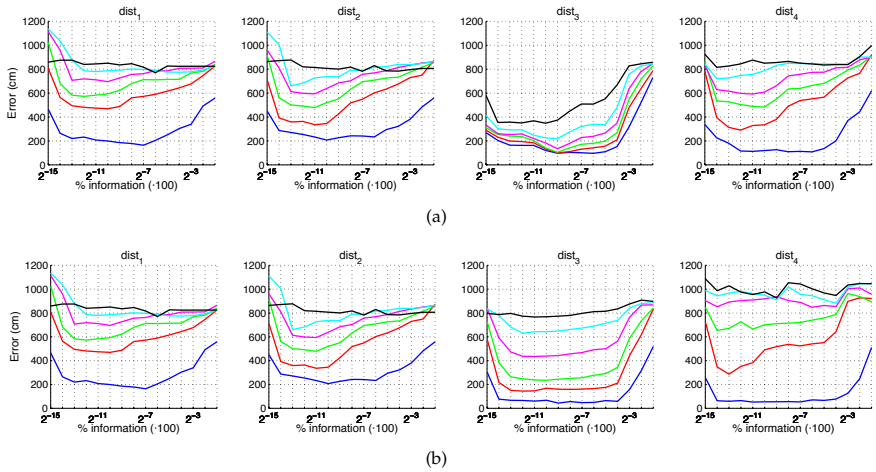


Figure 22. HOG average localization error with noise: (**a**) no filter and (**b**) homomorphic filter. Legend: — Original, — Noise 1, — Noise 2, — Noise 3, — Noise 4, — Noise 5.

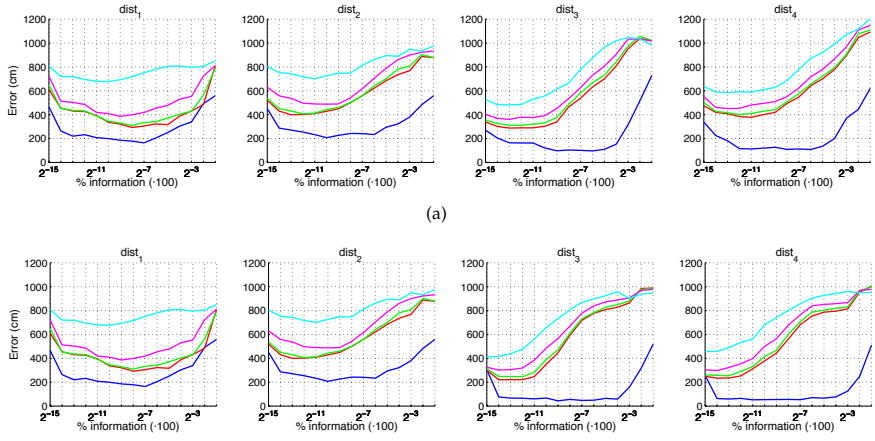


Figure 23. HOG average localization error with occlusions: (**a**) no filter and (**b**) homomorphic filter. Legend: — Original, — Occlusion 1, — Occlusion 2, — Occlusion 3, — Occlusion 4.

Additionally, Figures 24 and 25 present the results obtained with *gist* considering the presence of noise or occlusions, respectively, in the test images. In these figures, first, no filter is considered (a) and second, a homomorphic filtering is carried out both to the reference and the test images (b). Like in the case of FS, the percentages of information contained in each descriptor with respect to the information contained in each panoramic image can be obtained as $\left(\frac{2 \cdot k_7 \cdot m_1}{N_1 \cdot N_2} \cdot 100\right)$. The use of homomorphic filtering does not improve the localization results in any case. In the presence of noise, $dist_3$ presents the best results when considering an intermediate percentage of information.

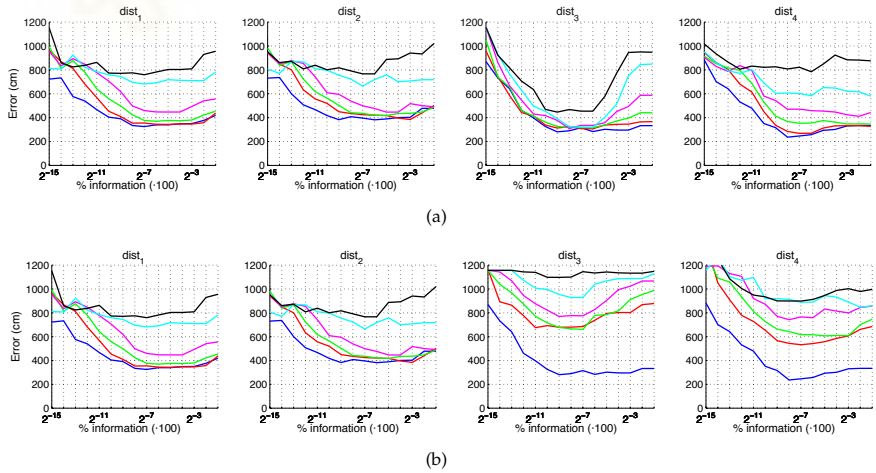


Figure 24. *Gist* average localization error with noise: (**a**) no filter and (**b**) homomorphic filtering. Legend: — Original, — Noise 1, — Noise 2, — Noise 3, — Noise 4, — Noise 5.

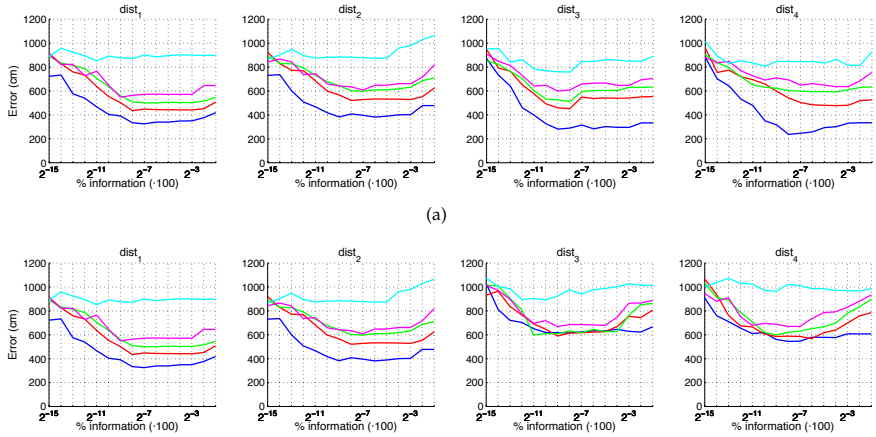


Figure 25. *Gist* average localization error with occlusions: (**a**) no filter and (**b**) homomorphic filtering. Legend: — Original, — Occlusion 1, — Occlusion 2, — Occlusion 3, — Occlusion 4.

Fourthly, Figures 26 and 27 present the results obtained with *Wi-SURF* considering the presence of noise or occlusions, respectively, in the test images. In these figures, first, no filter is considered (a) and second, a homomorphic filtering is carried out both to the reference and the test images (b). The information contained in each descriptor with respect to the information contained in each panoramic image can be obtained as $\left(\frac{k_9 \cdot w_1 \cdot 64}{N_1 \cdot N_2} \cdot 100\right)$. The use of homomorphic filtering does not reduce the localization error. In this case, the performance of the descriptor is severely influenced by the presence of noise. It is very significant that results without noise and occlusion are better than the errors obtained with the previous descriptors, but when these effects appear on the scene the results worsen sharply. In general, $dist_1$ and $dist_3$ present the best results when considering an intermediate or high percentage of information.

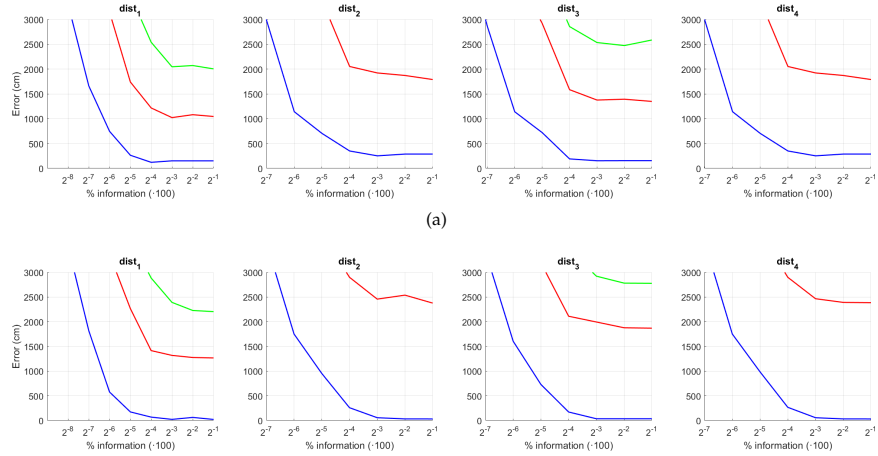


Figure 26. WS average localization error with noise: (**a**) no filter and (**b**) homomorphic filtering. Legend: — Original, — Noise 1, — Noise 2, — Noise 3, — Noise 4, — Noise 5.

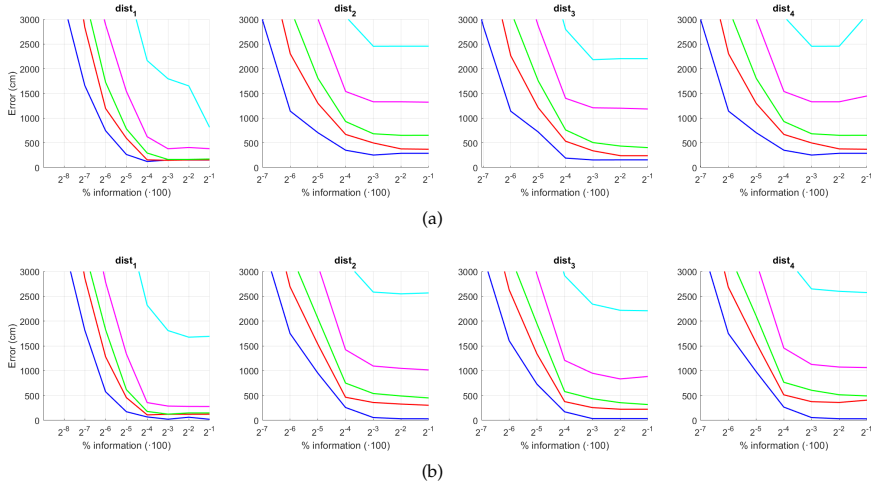


Figure 27. WS average localization error with occlusions: (**a**) no filter and (**b**) homomorphic filtering. Legend: — Original, — Occlusion 1, — Occlusion 2, — Occlusion 3, — Occlusion 4.

Figures 28 and 29 present the results obtained with the *BRIEF-gist* method considering the presence of noise or occlusions, respectively, in the test images. In these figures, first, no filter is considered (a) and second, a homomorphic filtering is carried out both to the reference and the test images (b). Like in previous figures, the percentages in the horizontal axis express the information contained in each descriptor with respect to the information contained in each panoramic image. In the case of BG, they can be obtained as $\left(\frac{k_{10} \cdot w_2}{N_1 \cdot N_2} \cdot 100\right)$. In this case, the best results are achieved with an intermediate amount of information, so it is not necessary to store a big quantity of information when building the descriptors. In addition, in general terms, the filter tends to improve the results. Comparing to the other description methods, *BRIEF-gist* presents higher error in ideal conditions, but it controls its error when noise appears on the scenes, obtaining good results even with high quantity of noise. Additionally it performs correctly when no occlusions take part on the image but it works wrongly when this phenomenon appears.

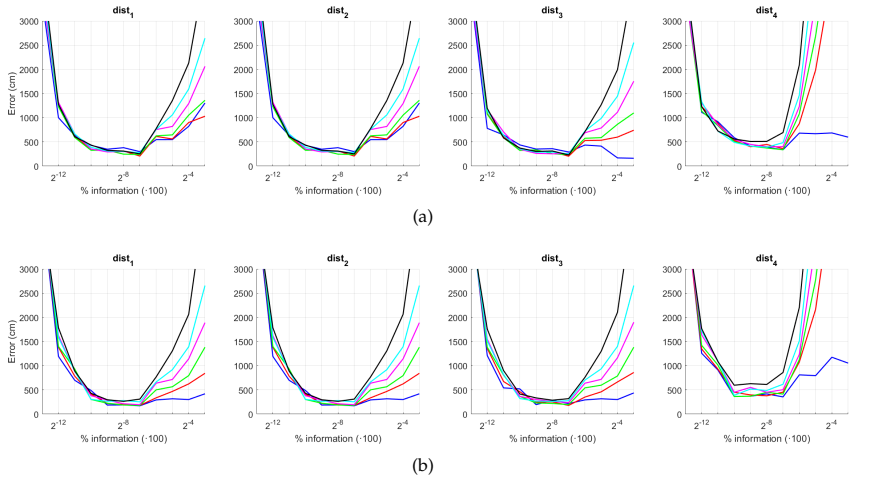


Figure 28. BG average localization error with noise: (**a**) no filter and (**b**) homomorphic filtering. Legend: — Original, — Noise 1, — Noise 2, — Noise 3, — Noise 4, — Noise 5.

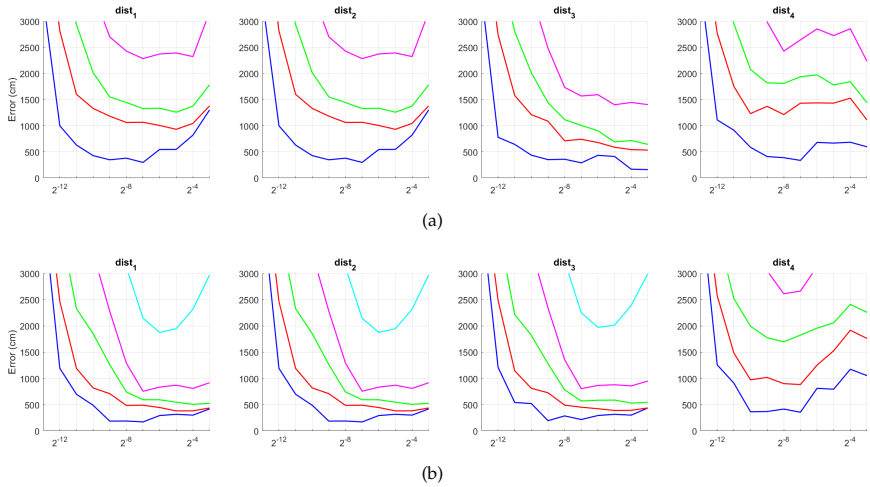


Figure 29. BG average localization error with occlusions: (**a**) no filter and (**b**) homomorphic filtering. Legend: — Original, — Occlusion 1, — Occlusion 2, — Occlusion 3, — Occlusion 4.

If we analyze jointly these results, we can arrive to some general conclusions. First, HOG presents very good localization results under ideal conditions. These results degrade in the presence of noise or occlusions, but some configurations resist these effects. Second, gist with no filter leads to worse results in ideal conditions, but it is robust against adverse effects, mainly against noise. Third, WS along with filter provides the best absolute localization results in ideal conditions. However, its performance sharply worsens with noise and occlusions. Finally, the results of BG in ideal conditions are not remarkable. However, this is the descriptor that presents more robustness in the presence of noise and occlusions, even in very unfavorable conditions.

5.3. Estimation of the Orientation

In this section, the problem of orientation estimation is addressed. To assess the performance of each description method in this task, independently of the results of the position estimation, the test image orientation descriptor is always compared with the orientation descriptor of the map image which was captured in the geometrically closest position. The problem is solved using the algorithms presented in Section 3, except those based on Radon transform, which proved to perform poorly in the image retrieval task.

First, the results obtained with the Fourier Signature are presented. Figure 30 shows the results of the orientation estimation. The influence of noise is also assessed in this figure. The results are expressed as average orientation error, in degrees, after repeating the experiment with the 1232 test images. This figure shows that the algorithm is very robust against the presence of noise. The optimal configuration is an intermediate to high number of rows (k_3) and an intermediate number of columns (k_4). A high number of columns worsens the results. Additionally, the presence of occlusions in the orientation estimation process is assessed in Figure 31. This figure shows that the influence of occlusions is higher, since the results tend to worsen as the level of occlusion increases. Nevertheless, some configurations of the parameters permit obtaining an average error lower than 10 deg even with 40% occlusions. The computational time of the orientation estimation process is shown in Figure 32, expressed in seconds. The descriptor based on FS is able to estimate the orientation relatively quickly for most configurations of k_3 and k_4 and only high values of both parameters produce a relatively high computation time.

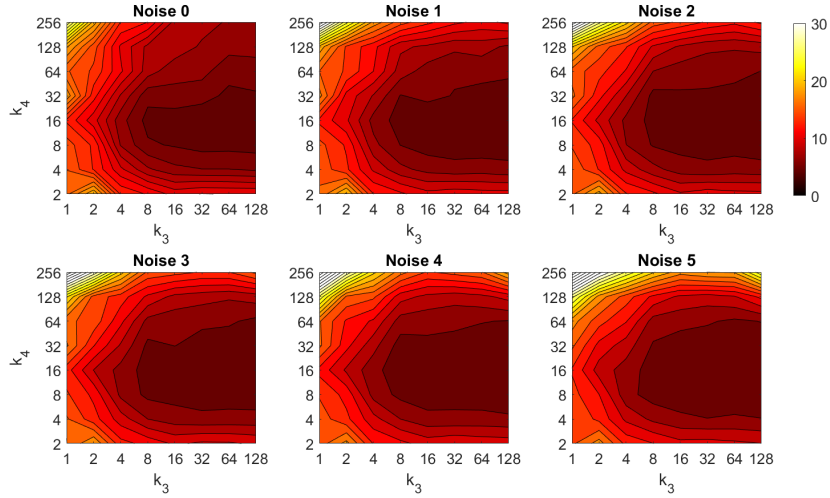


Figure 30. FS orientation estimation in the presence of noise. Average orientation error (deg).

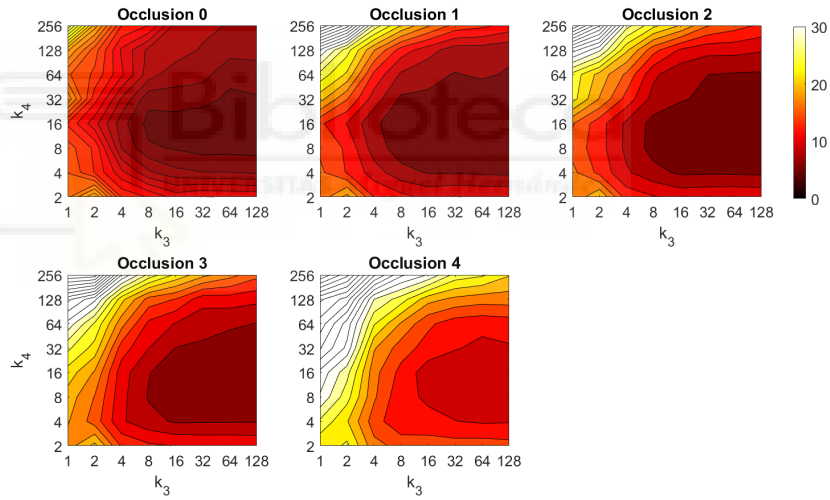


Figure 31. FS orientation estimation in the presence of occlusions. Average orientation error (deg).

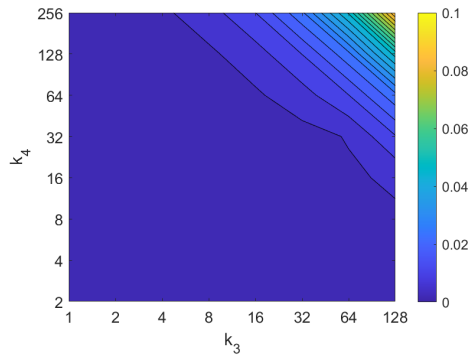


Figure 32. FS orientation estimation. Average computation time (s).

Second, the performance of the HOG descriptor is assessed, considering several values of the parameters l_1 (width of the vertical cells in the orientation descriptor) and d_1 (distance between consecutive vertical cells, which are overlapped). Figure 33 shows the average orientation error after considering all the test images. In addition, the influence of the presence of different levels of noise in the test images is analyzed. In general terms, low to intermediate values of d_1 and high values of l_1 produce the best results (lower orientation error). In addition, HOG proves to be a descriptor which is robust against the presence of noise, since the results do not change substantially as the level of noise increases. In general, HOG tends to present better results in orientation estimation comparing with FS. Furthermore, the influence of partial occlusions in orientation estimation is shown in Figure 34. As with FS, the influence of occlusions in the orientation estimation is substantial, and the results degrade quickly as the percentage of occlusions increases. Notwithstanding that, high values of l_1 tend to produce relatively low orientation error, independently of the level of occlusions. Finally, Figure 35 shows the necessary time to estimate the orientation (average time, expressed in seconds, after considering all the test images). Most configurations of l_1 and d_1 produce a relatively low computation time. Only very high values of l_1 combined with low values of d_1 output a substantially high calculation time.

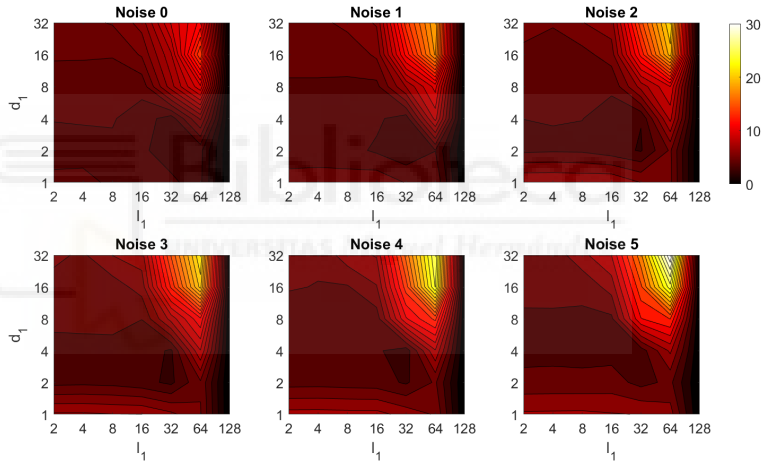


Figure 33. HOG orientation estimation in the presence of noise. Average orientation error (deg).

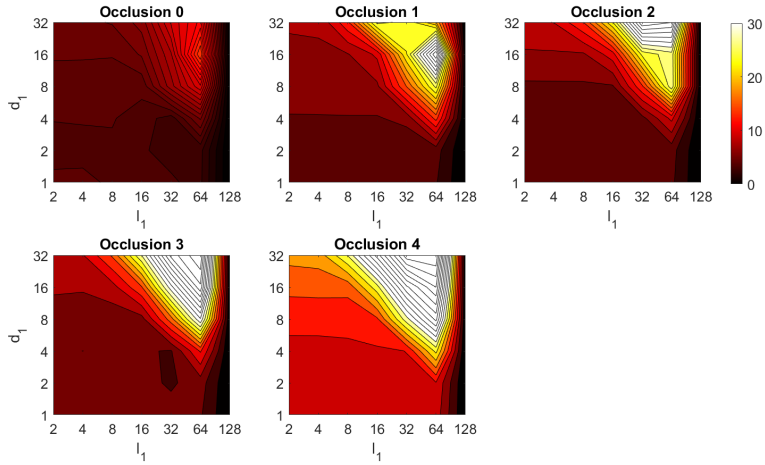


Figure 34. HOG orientation estimation in the presence of occlusion. Average orientation error (deg).

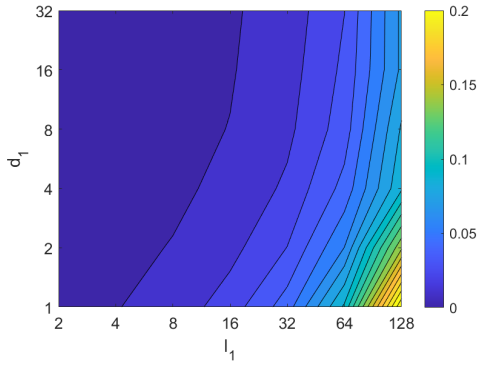


Figure 35. HOG orientation estimation. Average computation time (s).

Third, the results of relative orientation estimation with *gist* are presented and commented. Figure 36 shows the average orientation error (degrees) when considering different configurations of l_2 (width of the vertical blocks in the orientation descriptor) and d_2 (distance between two consecutive vertical blocks, which are overlapped). The influence of the level of occlusions is checked in this figure. In the case of this description method, the orientation error tends to increase as d_2 does. However, as in the case of HOG, high values of the width of the vertical blocks produce relatively good results independently of the value of d_2. To finish the experiments, the necessary time to estimate the orientation (average time in seconds) is shown in Figure 37. The figure shows that d_2 is the parameter that has a predominant influence upon the calculation time. Low values of this parameter produce a comparatively high computation time.

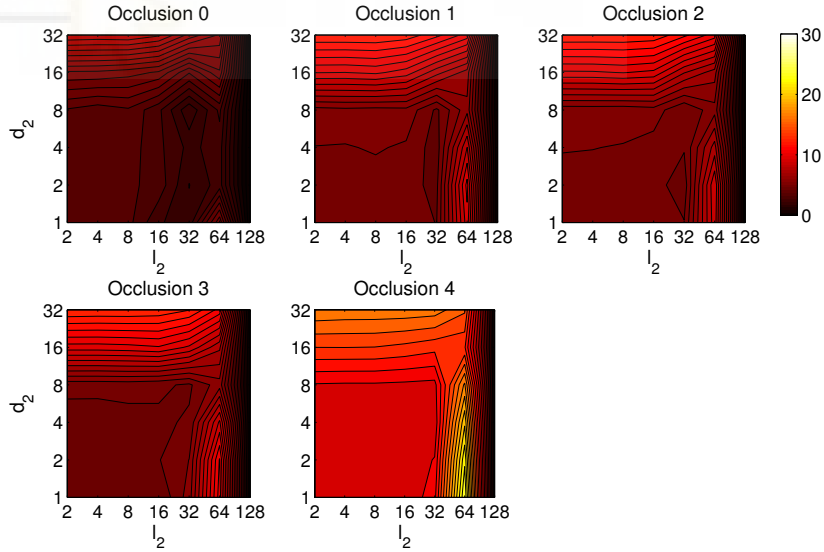


Figure 36. Gist orientation estimation in the presence of occlusion. Average orientation error (deg).

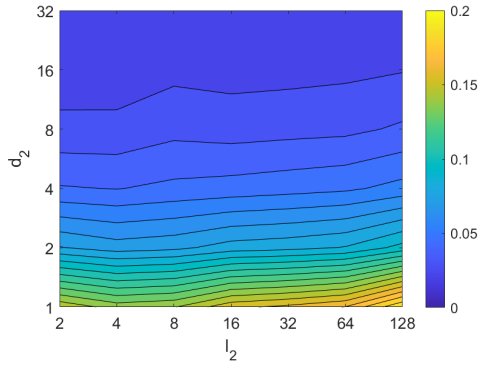


Figure 37. Gist orientation estimation. Average computation time (s).

In addition, the results of relative orientation estimation with *Wi-SURF* are presented and commented. Figure 38 shows the average orientation error (degrees) taking into account the noise influence considering the variation on the parameters k_9 and w_1. It shows a strong influence of the noise in the result. It is possible to check that results without noise are acceptable (about 5–10 deg), but the error increase considerably when the noise appears on the scenes. If the image is corrupted with noise with variance higher than $\sigma^2 = 0.0025$, the error is always more than 30 deg. The influence of the level of occlusions can be checked in the Figure 39. In the case of the occlusions, the results show more robustness, except for the results with 40% of occlusion that are considerably bad comparing with HOG. In general, the error tends to be optimized with middle values of w_1. To finish the experiments, the necessary time to estimate the orientation (average time in seconds) is shown in Figure 40. The figure shows that w_1 is the parameter that has a predominant influence upon the calculation time. High values of this parameter produce a comparatively high computation time.

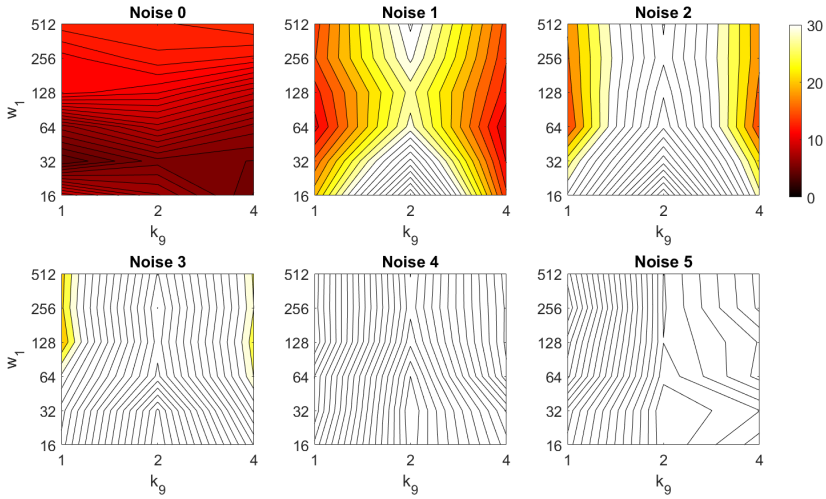


Figure 38. WS orientation estimation in the presence of noise. Average orientation error (deg).

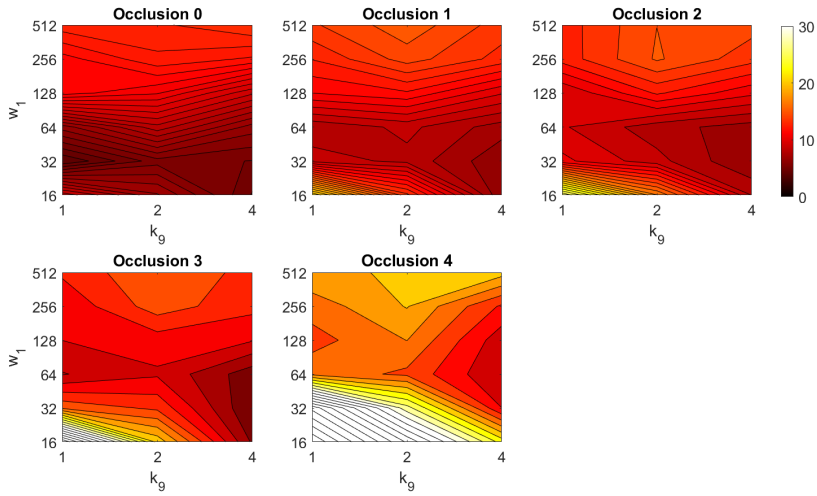


Figure 39. WS orientation estimation in the presence of occlusion. Average orientation error (deg).

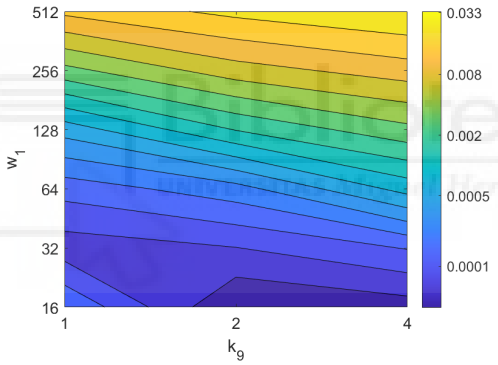


Figure 40. WS orientation estimation. Average computation time (s).

Additionally, the performance of the *BRIEF-gist* descriptor is assessed, considering several values of the parameters w_2 and k_{10}. Figure 41 shows the average orientation error after considering all the test images and the influence of the presence of different levels of noise. In general terms, the optimal configuration is an intermediate to high number of cells (k_{10}) and an intermediate number of windows (w_2). A high number of windows lead to worse results. In addition, *BRIEF-gist* proves to be a descriptor which is robust against the presence of noise, since the results do not change substantially as the level of noise increases. In general, *BRIEF-gist* tends to present better results in orientation estimation comparing with other descriptors. However, the influence of partial occlusions in orientation estimation has a worse influence, as shown in Figure 42. As with *Wi-SURF*, the algorithm performs considerably bad under the influence of occlusions. As before, intermediate values of w_2 output the best results. Finally, Figure 43 shows the necessary time to estimate the orientation. Most configurations of w_2 and k_{10} produce a relatively low computation time. Only very high values output a substantially high calculation time.

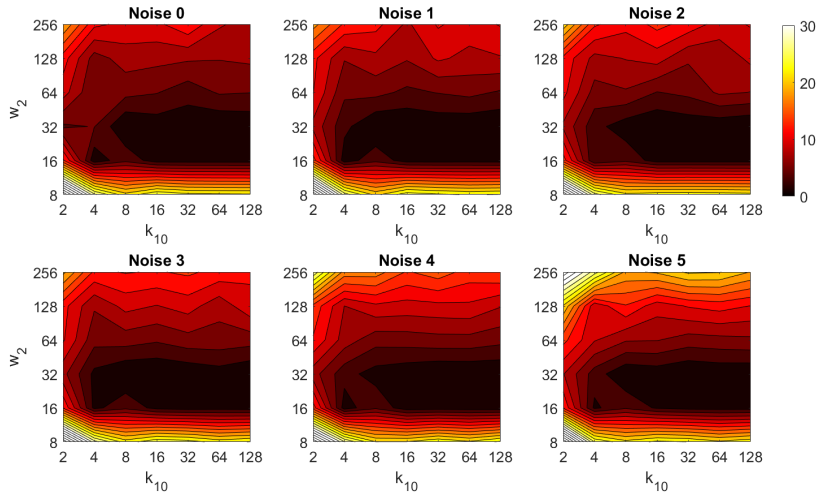


Figure 41. BG orientation estimation in the presence of noise. Average orientation error (deg).

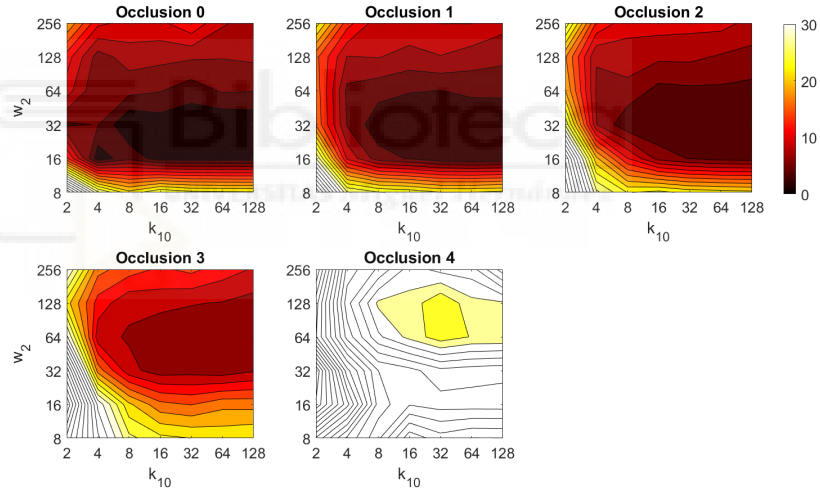


Figure 42. BG orientation estimation in the presence of occlusion. Average orientation error (deg).

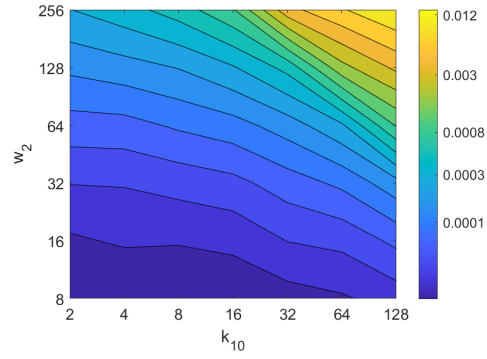


Figure 43. BG orientation estimation. Average computation time (s).

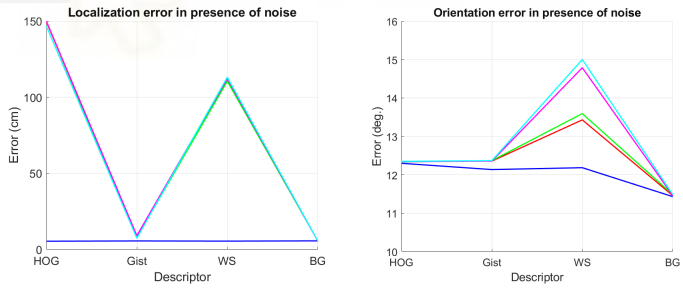
In general terms, HOG and *gist* produce relatively better results in the estimation of relative orientation, and the previous figures prove that it is possible to find some configurations of the most relevant parameters that offer a good balance between error and calculation time. Moreover, *Wi-SURF* and *BRIEF-gist* also offer acceptable errors in ideal conditions, and the calculation times are low. However, with these two descriptors, the orientation error tends to increase remarkably with the presence of occlusions and noise.

5.4. Evaluation with a Trajectory Dataset

To conclude the experimental section, a new experiment is carried out with a set of images extracted from the COLD dataset [72]. This publicly available dataset contains several sets of images that were captured while a mobile robot traversed a trajectory in some indoor environments. Therefore, the results in this section permit assessing the performance of the descriptors in a different environment and with a trajectory-like dataset.

To carry out the experiment, the Saarbrücken dataset is selected [72]. To create the training set, we have selected images from the Saarbrücken dataset in such a way that the distance between consecutive capture points is, on average, 30 cm. The rest of images are considered as test images, and they are used to solve the localization problem, as described in Section 3 .

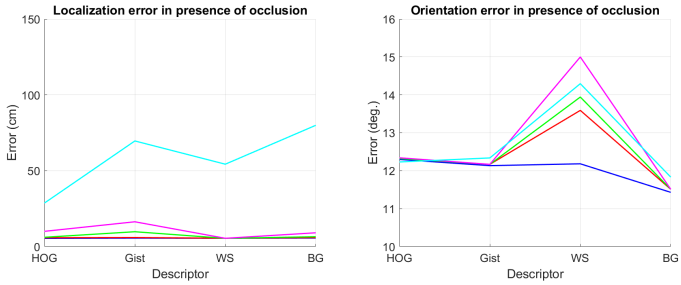
The results are presented in Figures 44 and 45. As in the previous experiments, we estimate both the position and the relative orientation of the robot and we consider either noise or occlusions in the test images. The descriptors included in this experiment are HOG, *gist*, WS and BG, since they have showed a good performance in the previous experiments. Additionally, their most relevant parameters are tuned with the values that provided, in general, best estimations in the previous subsections. The levels of noise or occlusion are the same than those included in the previous experiments: presence of different Gaussian noise ($\sigma^2 = \{0, 0.0025, 0.05, 0.01, 0.02\}$) and partial occlusions considering ($\{0, 5, 10, 20, 40\}$%) of the image occluded.



(a) (b)

Figure 44. Average errors with the COLD dataset in the presence of noise. (**a**) Average position error (cm) and (**b**) average orientation error (deg). Legend: — Original, — Noise 1, — Noise 2, — Noise 3, — Noise 4.

First, Figure 44 shows (a) the average error of the localization task (expressed in cm) and (b) the average error of the orientation retrieval task (expressed in deg). Several levels of noise are considered in this experiment. Second, Figure 45 shows the same results but considering several levels of partial occlusions. It is worth highlighting that these errors cannot be directly compared with the absolute errors presented in the previous subsections, since the experimental setup is different. Notwithstanding that, these figures permit assessing the relative performance of the descriptors with a trajectory-like dataset and knowing if the descriptors present similar tendencies in different kinds of environments and datasets.



(a) (b)

Figure 45. Average errors with the COLD dataset in the presence of occlusions. (**a**) Average position error (cm) and (**b**) average orientation error (deg). Legend : — Original, — Occlusion 1, — Occlusion 2, — Occlusion 3, — Occlusion 4.

Figure 44a shows that the relative performance of the descriptors when calculating the relative position in ideal conditions (i.e., with no noise) is quite similar. Additionally, *gist* and BG resist quite well the presence of noise. However, HOG and WS quickly degrade their performance as the level of noise increases. These results are in line with those presented in previous sections. About the relative orientation retrieval with noise, Figure 44b shows that HOG, *gist* and BG are quite robust, while WS performs worse with high levels of noise. Figure 45a proves that the four description methods present relatively good results in the presence of occlusions, except for the highest level of occlusion. In this case, HOG is the descriptor that best performs. About the orientation retrieval in presence of occlusions, Figure 45b shows that HOG, *gist* and BG perform well, independently on the level of occlusion, but WS quickly increases the error with high levels of occlusion.

6. Conclusions

This paper has focused on the study of the localization problem, using a previously built visual representation of the environment. The problem has been addressed as an absolute localization task, making use of the data provided by a catadioptric vision sensor mounted on the robot both to estimate both the position and the orientation of the robot. To extract relevant information from the images, methods based on the global appearance of the panoramic scenes have been implemented and assessed. A comparative evaluation has been carried out between six families of well-known global description methods.

The main contributions of the paper include an exhaustive study of global appearance techniques (FS, HOG, *gist*, WS, BG and RT) and the adaptation of some of these algorithms to store position and orientation information from panoramic scenes in such a way that both processes can be carried out sequentially. First, the position of the robot can be estimated and second, the orientation is estimated.

In addition, the computational cost to estimate the position and orientation has been studied, including the influence of the most relevant parameters. This study has revealed that FS and RT present a reasonable computational cost, and so do some specific configurations of HOG and *gist*, but *Wi-SURF* and *BRIEF-gist* are less competitive as far as computation time is concerned. From this point of view, FS, RT, HOG and *gist* could be feasible in real time applications. In addition to this, the performance of the descriptors has been tested in localization tasks. First, we have focused on the image retrieval problem. All the description methods have been tested along with several distance measures, and the results have shown that *Wi-SURF* and *BRIEF-gist* present the best relative results. Additionally, HOG with certain distance measures present very good results and the best relation between computational time and image retrieval rate. Second, the relative error of the position estimation has been studied. It has corroborated that: (a) HOG presents very good localization results under ideal conditions and is quite robust to noise and occlusions,

(b) Wi-SURF provides the most competitive results under ideal conditions but is very negatively influenced by noise and occlusions and (c) BRIEF-gist is very robust against these effects, but its results in ideal conditions are not remarkable. To finish, the problem of orientation estimation has been addressed. The best results are obtained with WS and BG but only when there is neither noise nor occlusions. If these phenomena are present, HOG and *gist* perform more robustly.

These results have demonstrated that global-appearance methods are a feasible approach to solve the localization task. Thanks to them, the robot can build a model of the environment and use it to estimate with accuracy the position and orientation of the robot in the environment, with computational efficiency. This fact may have interesting implications in future developments in the field of mobile robotics. As an example, this concept can be used to build hybrid maps that arrange the information in several layers, with different accuracy: a high level layer that permits carrying out a rough and quick localization and a lower layer that contains information with geometric accuracy and allows the robot to refine the estimation of its position. Global-appearance methods can be used on their own or in conjunction with feature-based techniques to develop algorithms that face these problems efficiently.

All these facts encourage us to go into this framework in depth. To build a fully autonomous mapping and localization system, several future works should be considered. First, the image collection process could be automated to obtain an optimal representation of the environment. Second, the mapping and localization processes could be integrated in a topological SLAM system that carries out both the model creation and the localization from the scratch. To optimize these algorithms we also consider carrying out a complete comparison between global-appearance and feature-based techniques as a future work.

Author Contributions: Conceptualization, L.P. and O.R.; methodology, L.P., O.R. and A.P.; software, L.P., M.B. and V.R.; validation, L.P., M.B. and V.R.; formal analysis, O.R. and A.P.; investigation, A.P., M.B. and V.R.; resources, L.P. and O.R.; data curation, M.B. and V.R.; writing—original draft preparation, L.P. and O.R.; writing—review and editing, V.R. and O.R.; visualization, A.P., M.B. and V.R.; supervision, L.P. and O.R.; project administration, L.P. and O.R.; funding acquisition, L.P. and O.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Spanish Government through the project DPI2016-78361-R (AEI/FEDER, UE): *Creación de mapas mediante métodos de apariencia visual para la navegación de robots*, by the Generalitat Valenciana through the project AICO/2019/031: *Creación de modelos jerárquicos y localización robusta de robots móviles en entornos sociales* and by the Generalitat Valenciana and the FSE through the grant ACIF/2018/224.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://arvc.umh.es/db/images/quorumv/ (accessed on 9 May 2021).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Reinoso, O.; Payá, L. Special Issue on Mobile Robots Navigation. *Appl. Sci.* **2020**, *10*, 1317. [CrossRef]
2. Reinoso, O.; Payá, L. Special Issue on Visual Sensors. *Sensors* **2020**, *20*, 910. [CrossRef] [PubMed]
3. Junior, J.M.; Tommaselli, A.; Moraes, M. Calibration of a catadioptric omnidirectional vision system with conic mirror. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 97–105. [CrossRef]
4. Coors, B.; Paul Condurache, A.; Geiger, A. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 518–533.

5. Sun, C.; Hsiao, C.W.; Sun, M.; Chen, H.T. HorizonNet: Learning Room Layout With 1D Representation and Pano Stretch Data Augmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
6. Pintore, G.; Agus, M.; Gobbetti, E. AtlantaNet: Inferring the 3D Indoor Layout from a Single 360 Image Beyond the Manhattan World Assumption. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 432–448.
7. Xu, S.; Chou, W.; Dong, H. A Robust Indoor Localization System Integrating Visual Localization Aided by CNN-Based Image Retrieval with Monte Carlo Localization. *Sensors* **2019**, *19*, 249. [CrossRef]
8. Leyva-Vallina, M.; Strisciuglio, N.; Lopez-Antequera, M.; Tylecek, R.; Blaich, M.; Petkov, N. TB-Places: A Data Set for Visual Place Recognition in Garden Environments. *IEEE Access* **2019**, *7*, 52277–52287. [CrossRef]
9. Cebollada, S.; Payá, L.; Flores, M.; Peidró, A.; Reinoso, O. A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. *Expert Syst. Appl.* **2020**, *167*, 114195. [CrossRef]
10. Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
11. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
12. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary robust independent elementary features. In Proceedings of the European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010; pp. 778–792.
13. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary robust invariant scalable keypoints. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
14. Rublee, E.; Rabud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 ICCV 2011: International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
15. Alahi, A.; Ortiz, R.; Vandergheynst, P. FREAK: Fast Retina Keypoint. In Proceedings of the CVPR 2012: Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 510–517.
16. Yang, X.; Cheng, K.T.T. Local difference binary for ultrafast and distinctive feature description. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 188–194. [CrossRef]
17. Krose, B.; Bunschoten, R.; Hagen, S.; Terwijn, B.; Vlassis, N. Visual homing in enviroments with anisotropic landmark distrubution. *Auton. Robot.* **2007**, *23*, 231–245.
18. Menegatti, E.; Maeda, T.; Ishiguro, H. Image-based memory for robot navigation using properties of omnidirectional images. *Robot. Auton. Syst.* **2004**, *47*, 251–267. [CrossRef]
19. Oliva, A.; Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
20. Ulrich, I.; Nourbakhsh, I. Appearance-based place recognition for topological localization. In Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24–28 April 2000; pp. 1023–1029.
21. Amorós, F.; Payá, L.; Mayol-Cuevas, W.; Jiménez, L.M.; Reinoso, O. Holistic Descriptors of Omnidirectional Color Images and Their Performance in Estimation of Position and Orientation. *IEEE Access* **2020**, *8*, 81822–81848. [CrossRef]
22. Milford, M. Visual Route Recognition with a Handful of Bits. In Proceedings of the Robotics: Science and Systems, Sydney, NSW, Australia, 9–13 July 2012.
23. Berenguer, Y.; Payá, L.; Valiente, D.; Peidró, A.; Reinoso, O. Relative Altitude Estimation Using Omnidirectional Imaging and Holistic Descriptors. *Remote Sens.* **2019**, *11*, 323. [CrossRef]
24. Yuan, X.; Martínez-Ortega, J.F.; Fernández, J.A.S.; Eckert, M. AEKF-SLAM: A new algorithm for robotic underwater navigation. *Sensors* **2017**, *17*, 1174. [CrossRef] [PubMed]
25. Luthardt, S.; Willert, V.; Adamy, J. LLama-SLAM: Learning high-quality visual landmarks for long-term mapping and localization. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2645–2652.
26. Cao, L.; Ling, J.; Xiao, X. Study on the Influence of Image Noise on Monocular Feature-Based Visual SLAM Based on FFDNet. *Sensors* **2020**, *20*, 4922. [CrossRef] [PubMed]
27. Shamsfakhr, F.; Bigham, B.S.; Mohammadi, A. Indoor mobile robot localization in dynamic and cluttered environments using artificial landmarks. *Eng. Comput.* **2019**, *36*, 400–419. [CrossRef]
28. Lin, J.; Peng, J.; Hu, Z.; Xie, X.; Peng, R. ORB-SLAM, IMU and Wheel Odometry Fusion for Indoor Mobile Robot Localization and Navigation. *Acad. J. Comput. Inf. Sci.* **2020**, *3*. [CrossRef]
29. Gil, A.; Mozos, O.M.; Ballesta, M.; Reinoso, O. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Mach. Vis. Appl.* **2010**, *21*, 905–920. [CrossRef]
30. Dong, X.; Dong, X.; Dong, J.; Zhou, H. Monocular Visual-IMU Odometry: A Comparative Evaluation of Detector–Descriptor-Based Methods. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 2471–2484. [CrossRef]
31. Menegatti, E.; Zocaratto, M.; Pagello, E.; Ishiguro, H. Image-based Monte Carlo Localisation with Omnidirectional Images. *Robot. Auton. Syst.* **2004**, *48*, 17–30. [CrossRef]
32. Murillo, A.; Guerrero, J.; Sagües, C.; Filliat, D. Surf features for efficient robot localization with omnidirectional images. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy, 10–14 April 2007; pp. 3901–3907.

33. Siagian, C.; Itti, L. Biologically Inspired Mobile Robot Vision Localization. *IEEE Trans. Robot.* **2009**, *25*, 861–873. [CrossRef]
34. Payá, L.; Amorós, F.; Fernández, L.; Reinoso, O. Performance of Global-Appearance Descriptors in Map Building and Localization Using Omnidirectional Vision. *Sensors* **2014**, *14*, 3033–3064. [CrossRef]
35. Khaliq, A.; Ehsan, S.; Chen, Z.; Milford, M.; McDonald-Maier, K. A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes. *IEEE Trans. Robot.* **2019**, *36*, 561–569. [CrossRef]
36. Román, V.; Payá, L.; Cebollada, S.; Reinoso, Ó. Creating Incremental Models of Indoor Environments through Omnidirectional Imaging. *Appl. Sci.* **2020**, *10*, 6480. [CrossRef]
37. Marinho, L.B.; Almeida, J.S.; Souza, J.W.M.; Albuquerque, V.H.C.; Rebouças Filho, P.P. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Syst. Appl.* **2017**, *72*, 1–17. [CrossRef]
38. Ma, J.; Zhao, J. Robust topological navigation via convolutional neural network feature and sharpness measure. *IEEE Access* **2017**, *5*, 20707–20715. [CrossRef]
39. Paya, L.; Reinoso, O.; Berenguer, Y.; Ubeda, D. Using omnidirectional vision to create a model of the environment: A comparative evaluation of global appearance descriptors. *J. Sens.* **2016**, *2016*, 1–21. [CrossRef]
40. Arroyo, R.; Alcantarilla, P.F.; Bergasa, L.M.; Yebes, J.J.; Bronte, S. Fast and effective visual place recognition using binary codes and disparity information. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 3089–3094.
41. Berenguer, Y.; Payá, L.; Peidró, A.; Gil, A.; Reinoso, O. Nearest Position Estimation Using Omnidirectional Images and Global Appearance Descriptors. In *Robot 2015: Second Iberian Robotics Conference*; Springer: Cham, Switzerland, 2016; pp. 517–529.
42. Ishiguro, H.; Tsuji, S. Image-based memory of environment. In Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96 (IROS 96), Osaka, Japan, 4–8 November 1996; Volume 2, pp. 634–639. [CrossRef]
43. Sturzl, W.; Mallot, H. Efficient visual homing based on Fourier transformed panoramic images. *Robot. Auton. Syst.* **2006**, *54*, 300–313. [CrossRef]
44. Horst, M.; Möller, R. Visual place recognition for autonomous mobile robots. *Robotics* **2017**, *6*, 9. [CrossRef]
45. Dalal, N.; Triggs, B. Histograms of Oriented Gradients fot Human Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume II, pp. 886–893.
46. Zhu, Q.; Avidan, S.; Yeh, M.C.; Cheng, K.T. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1491–1498. [CrossRef]
47. Hofmeister, M.; Liebsch, M.; Zell, A. Visual self-localization for small mobile robots with weighted gradient orientation histograms. In Proceedings of the 40th International Symposium on Robotics, Barcelona, Spain, 10–13 March 2009; IFR: Frankfurt am Main, Germany, 2009; pp. 87–91.
48. Hofmeister, M.; Vorst, P.; Zell, A. A comparison of Efficient Global Image Features for Localizing Small Mobile Robots. In Proceedings of the 41st International Symposium on Robotics, Munich, Germany, 7–9 June 2010; pp. 143–150.
49. Aslan, M.F.; Durdu, A.; Sabanci, K.; Mutluer, M.A. CNN and HOG based comparison study for complete occlusion handling in human tracking. *Measurement* **2020**, *158*, 107704. [CrossRef]
50. Neumann, D.; Langner, T.; Ulbrich, F.; Spitta, D.; Goehring, D. Online vehicle detection using Haar-like, LBP and HOG feature based image classifiers with stereo vision preselection. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 773–778.
51. Payá, L.; Fernández, L.; Gil, A.; Reinoso, O. Map Building and Monte Carlo Localization Using Global Appearance of Omnidirectional Images. *Sensors* **2010**, *10*, 11468–11497. [CrossRef]
52. Oliva, A.; Torralba, A. Building the gist of ascene: The role of global image features in recognition. *Prog. Brain Res. Spec. Issue Vis. Percept.* **2006**, *155*, 23–36.
53. Torralba, A. Contextual priming for object detection. *Int. J. Comput. Vis.* **2003**, *53*, 169–191. [CrossRef]
54. Siagian, C.; Itti, L. Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 300–312. [CrossRef]
55. Chang, C.K.; Siagian, C.; Itti, L. Mobile robot vision navigation and localization using Gist and Saliency. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 4147–4154. [CrossRef]
56. Murillo, A.; Singh, G.; Kosecka, J.; Guerrero, J. Localization in Urban Environments Using a Panoramic Gist Descriptor. *IEEE Trans. Robot.* **2013**, *29*, 146–160. [CrossRef]
57. Liu, Y.; Zhang, H. Visual loop closure detection with a compact image descriptor. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligen Robots and Systems (IROS), Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 1051–1056.
58. Su, Z.; Zhou, X.; Cheng, T.; Zhang, H.; Xu, B.; Chen, W. Global localization of a mobile robot using lidar and visual features. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, Macao, 5–8 December 2017; pp. 2377–2383.

59. Andreasson, H.; Treptow, A.; Duckett, T. Localization for mobile robots using panoramic vision, local features and particle filter. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 3348–3353.
60. Agrawal, M.; Konolige, K.; Blas, M.R. Censure: Center surround extremas for realtime feature detection and matching. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 102–115.
61. Badino, H.; Huber, D.; Kanade, T. Real-time topometric localization. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1635–1642.
62. Zhang, M.; Han, S.; Wang, S.; Liu, X.; Hu, M.; Zhao, J. Stereo Visual Inertial Mapping Algorithm for Autonomous Mobile Robot. In Proceedings of the 2020 3rd International Conference on Intelligent Robotic and Control Engineering (IRCE), Oxford, UK, 10–12 August 2020; pp. 97–104.
63. Aladem, M.; Rawashdeh, S.A. Lightweight visual odometry for autonomous mobile robots. *Sensors* **2018**, *18*, 2837.
64. Sünderhauf, N.; Protzel, P. Brief-gist-closing the loop by simple means. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 1234–1241.
65. Radon, J. 1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Class. Pap. Mod. Diagn. Radiol.* **2005**, *5*, 21.
66. Hoang, T.V.; Tabbone, S. A geometric invariant shape descriptor based on the Radon, Fourier, and Mellin transforms. In Proceedings of the 2010 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010; pp. 2085–2088.
67. Hasegawa, M.; Tabbone, S. A shape descriptor combining logarithmic-scale histogram of radon transform and phase-only correlation function. In Proceedings of the 2011 International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011; pp. 182–186.
68. Berenguer, Y.; Payá, L.; Ballesta, M.; Reinoso, O. Position estimation and local mapping using omnidirectional images and global appearance descriptors. *Sensors* **2015**, *15*, 26368–26395. [PubMed]
69. Juliá, M.; Gil, A.; Reinoso, O. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Auton. Robot.* **2012**, *33*, 427–444. [CrossRef]
70. Liu, S.; Li, S.; Pang, L.; Hu, J.; Chen, H.; Zhang, X. Autonomous Exploration and Map Construction of a Mobile Robot Based on the TGHM Algorithm. *Sensors* **2020**, *20*, 490. [CrossRef] [PubMed]
71. ARVC. Automation, Robotics and Computer Vision Research Group. Miguel Hernández University. Spain. Quorum 5 Set of Images. Available online: http://arvc.umh.es/db/images/quorumv/ (accessed on 29 December 2020).
72. Pronobis, A.; Caputo, B. COLD: COsy Localization Database. *Int. J. Robot. Res. (IJRR)* **2009**, *28*, 588–594. [CrossRef]

applied sciences

MDPI

Article

Creating Incremental Models of Indoor Environments through Omnidirectional Imaging

Vicente Román *,†,‡, Luis Payá ‡, Sergio Cebollada ‡ and Óscar Reinoso ‡

Department of Systems Engineering and Automation, Miguel Hernández University, 03202 Elche, Spain; lpaya@umh.es (L.P.); sergio.cebollada@Umh.es (S.C.); o.reinoso@umh.es (Ó.R.)

* Correspondence: v.roman@umh.es; Tel.: +34-96-665-2435

† Current address: Avda. de la Universidad, s/n. Ed, Innova, 03202 Elche, Spain.

‡ These authors contributed equally to this work.

Received: 3 August 2020; Accepted: 13 September 2020; Published: 17 September 2020

Abstract: In this work, an incremental clustering approach to obtain compact hierarchical models of an environment is developed and evaluated. This process is performed using an omnidirectional vision sensor as the only source of information. The method is structured in two loop closure levels. First, the Node Level Loop Closure process selects the candidate nodes with which the new image can close the loop. Second, the Image Level Loop Closure process detects the most similar image and the node with which the current image closed the loop. The algorithm is based on an incremental clustering framework and leads to a topological model where the images of each zone tend to be clustered in different nodes. In addition, the method evaluates when two nodes are similar and they can be merged in a unique node or when a group of connected images are different enough to the others and they should constitute a new node. To perform the process, omnidirectional images are described with global appearance techniques in order to obtain robust descriptors. The use of such technique in mapping and localization algorithms is less extended than local features description, so this work also evaluates the efficiency in clustering and mapping techniques. The proposed framework is tested with three different public datasets, captured by an omnidirectional vision system mounted on a robot while it traversed three different buildings. This framework is able to build the model incrementally, while the robot explores an unknown environment. Some relevant parameters of the algorithm adapt their value as the robot captures new visual information to fully exploit the features' space, and the model is updated and/or modified as a consequence. The experimental section shows the robustness and efficiency of the method, comparing it with a batch spectral clustering algorithm.

Keywords: mapping; incremental clustering; mobile robots; global-appearance descriptors; omnidirectional images; computer vision

1. Introduction

Presently, the use of visual information in mobile robotics is widely expanded. Independent of the final task it was designed for, an autonomous mobile robot must solve continuously two crucial problems: it has to build a model of the environment (mapping) and to estimate the position of the robot within this model (localization). Both critical problems should be solved with an acceptable accuracy and computational cost.

Mapping was facilitated by the continuous progress of mobile robots' abilities in perception and computation, which enable them to improve their operability in large and heterogeneous zones without the necessity of introducing changes into the environment structure. Two main frameworks were used in order to carry out the mapping task: the metric maps [1]; and the topological maps [2].

Regarding the topological maps, some related works propose arranging the information hierarchically, into several layers with different levels of granularity [3,4].

Hierarchical maps constitute a convenient framework to arrange the information. Notwithstanding, visual mapping remains an important research field in robotics due to the problems that the algorithms may have in heterogeneous and challenging environments. An important alternative to build hierarchical maps is by using some clustering techniques, such as spectral clustering [5]. Some researchers used spectral clustering methods along with visual information to build topological maps [6–9]. Spectral clustering proved to be useful because it can cluster robustly highly dimensional information compared to other well-known methods [10,11].

Even though spectral clustering was used to build maps in mobile robotics with good results [11], the problems of using the standard spectral method in incremental mapping are threefold. Firstly, most spectral clustering methods require the number of final nodes to be indicated previously. Secondly, computing similarities among entities can be hard in terms of computational cost. This is especially noticeable when the data set becomes large. Finally, as a consequence of the previous problem, it is not possible to perform spectral clustering *on-line* when the dataset is constantly growing (i.e., The robot is continuously moving and capturing new images which must be included in the model).

Presently, incremental mapping is a key ability because it would enable mobile robots to gradually build or update a model as they explore the initially unknown environment and capture new information. For this purpose, incremental clustering methods can be useful [12].

In this work, we propose a framework based on incremental clustering with the objective of creating a topological hierarchical map incrementally, using only visual information. Each group of similar images will be included in a cluster with a representative descriptor associated to it. To avoid the necessity of setting the number of clusters beforehand, our proposal defines a set of adaptive thresholds. Depending on them, the algorithm behaves more or less restrictively and therefore creates a higher or lower number of clusters as it progresses. In addition, the method proposed in this paper is able to perform *on-line*, updating the model every time the robot captures a new image.

When using computer vision to build a model of the environment, it is important to verify the performance of the method when the visual appearance of the environment changes substantially. In real-operating conditions the robot has to cope with different events: different lighting conditions during the operation, scenes partially occluded by people or other mobile robots and changes in the scene due, for example to the furniture position. For that reason experiments were carried out in a real environment during working hours.

The remainder of the paper is structured as follows. In the following section, related works on visual information and incremental and hierarchical mapping are outlined. In Section 3, global appearance descriptors are defined. The mapping method is presented in Section 4. In Section 5, relevant information about the datasets, tools and acquisition equipment used in the experiments is presented. In Section 6, the results of the experiments are shown. Finally, conclusions and future research lines are presented in Section 7. An extra section named 'Supplementary Materials' includes some tables of symbols, which summarize all the variables and parameters used to describe the proposed method.

2. Related Works

2.1. Image Description

As explained in the introduction, mobile robots have to solve the mapping and localization problems during their autonomous navigation. To deal with these tasks the robot has to capture relevant information about its movement and the environment, and a variety of tools can be found in the literature to obtain that information and to process it. Encoders permit calculating the odometry of the robot and they are one of the main sensors used in mobile robotics, but the information obtained from the encoders should not be used as an absolute measure because of the error they accumulate. Taking that into

account, in real implementations, odometry data are complemented by other sensors. For instance, it can be complemented with sonars [13] or lasers [14]. In addition, mobile robots can work also with GPS, which presents a good performance in outdoor applications but it has little ability to offer signal in indoor or narrow outdoor zones. Complementing these sensors, visual sensors constitute currently the principal area to investigate new approaches to robot navigation. Cameras offer a lot of information from the environment with a relatively low cost, they can be used in outdoor and indoor environments and they permit solving other high-level tasks such as human identification, face recognition [15] and obstacles detection [16]. There are some examples of sensors combinations: For example Choi et al. [17] present a system that combines sonar with visual information to perform SLAM (Simultaneous Localization And Mapping) tasks. More information about these systems can be found in [18], where a review of these techniques is done. The visual information can be captured with a variety of devices; from a single camera [19], stereo cameras [20] or omnidirectional cameras [21,22].

Solving navigation tasks using only visual information is a challenging problem. Images contain much information and relevant data should be extracted from them in order to ease the mapping and localization tasks. The use of local descriptors was the classical approach for obtaining relevant information from the images and the method consists of extracting outstanding landmarks or regions and describing them using algorithms such as SIFT (Scale-Invariant Feature Transform) [23], SURF (Speeded-Up Robust Features) [24] or BRIEF (Binary Robust Independent Elementary Features) [25]. This is a mature alternative and many researchers make use of it in mapping and localization. For example, Angeli et al. [26] proposed using these descriptors to perform topological localization. Valgren and Lilienthal [7] did that work in outdoor areas. Murillo et al. [27] present a comparative study for the localization task using local-appearance descriptors in indoor environments. A comparative evaluation of this kind of local appearance methods was made by Gil et al. [28]. They evaluated the repeatability, the invariance against small changes and distinctiveness of the descriptors under different perceptual conditions. They proved that the behaviour of local appearance descriptors can deteriorate when unexpected conditions appear. Taking it into account that the information not only changes while the robot is moving and the perspective varies, but also due to changing lighting conditions, occlusions by furniture or people and noise during the acquisition, it is necessary to calculate more robust and invariant descriptors. In this sense, global-appearance descriptors constitute a powerful alternative, which consists of describing each image as a whole, without detecting any landmark or local feature. This method creates a more intuitive representation of the environment, simplifies the navigation process and permits building topological maps of the environment. Since each image is described with a unique vector, localization can be carried out with simple algorithms, based on the pairwise comparison of vectors. Additionally, calculating global-appearance descriptors from omnidirectional images constitutes a powerful approach due to the wide field of view of such images, leading to robust and rotationally invariant descriptors, such as those presented in the next paragraph.

Diverse global appearance methods were studied in recent years. One of these methods is based on the Histogram of Oriented Gradients (HOG). Using this descriptor, diverse problems were solved. For instance Dalal and Triggs [29] explain how to use it to detect pedestrians in a real situation, Paya et al. [30,31] use HOG to solve the localization problems and to create hierarchical maps. Gist is another description method, proposed by Oliva and Torralba in [32] and it was used in works such as [33] where it is described as a biologically inspired vision localization system and the descriptor is tested in different outdoor environments. It was also used by Zhou et al. [34] to solve the localization through matching the robot's current view with the best key-frame in the database. In addition, some other works define other global appearance description methods such as the Discrete Fourier Transform [35], the alternative used by Paya et al. [30] to perform map creation tasks, or Radon Transform [36], used in [37] to find the nearest neighbour in a dense map previously created. Román et al. [38] develop a comparison among these global appearance descriptors performing a mobile robot localization in a real environment under changing lighting conditions [38]. In addition, more recently, deep learning techniques are studied with the idea of creating new global

appearance descriptors. For example, Xu et al. [39] and Leyva et al. [40] proposed holistic descriptors based on a CNN (Convolutional Neural Network) to obtain the most probable robot position and Cebollada et al. [11] carry out a comparison of analytic global-appearance descriptors and CNN-based descriptors in localization tasks.

2.2. Mapping and Clustering Methods

Another important research topic in mobile robotics is map creation. In the related literature, two main different frameworks were proposed in order to obtain maps in mobile robotics. First, metric maps, which represent the environment with geometric accuracy. Second, topological maps, which describe the environment as a graph containing a set of locations with the related links among them, with no metric information. Regarding these options, some authors proposed storing information in the map hierarchically, into a set of layers that contain information from the environment with different levels of granularity. Such arrangement permits solving localization efficiently, in two phases. First, a rough, but fast localization is carried out using the high-level layers; second, a fine localization is performed in a smaller area using the low-level layers. Therefore hierarchical maps constitute an efficient alternative to build maps with autonomous mobile robots [41–43].

Focusing on hierarchical mapping, Balaska et al. [44] develop an unsupervised semantic mapping and propose a localization method. SURF points are used to carry out the clustering and the map is corrected by means of odometry. Korrapati and Mezouar [45] perform clustering and propose image loop closure with the objective of building hierarchical maps. They work with omnidirectional images and local features. Kostavelis et al. [4] propose an augmented navigation graph, an extreme hierarchical map that consist of 4 layers. On the lowest layer, metric information is stored, so it is easier for the robot to navigate and perform localization tasks. As the level of the layer increases so the abstraction level does, in such a way that the highest-level layer is a graph with a conceptual representation of detected places. Clusters on that layer are connected with an indicator that represents probability of success to make a transition to the adjacent cluster.

Additionally, loop closure detection constitutes a crucial step when designing a method to create accurate maps, as shown in [46]. In this work, when a new closure is detected, the new image is stored as a combination of the previous images instead of as a new image. In [47], loop closure detection is performed in two phases using the information of a partially built map: In the first phase, global descriptors are used to find loop closure candidates. In the second one, the loop is closed by choosing the best result among the candidates, using local features. As far as loop closure detection is concerned, the difference of our proposal consists of performing this detection taking advantage of the hierarchical structure of the map.

Finally, as stated in the introduction, we propose building hierarchical maps incrementally. Some authors addressed previously this problem by means of incremental clustering. In [12], a method to build topological maps incrementally is proposed, using the SIFT descriptor, and the number of clusters continuously increases while the robot is navigating. The result is a well-separated clustered route, but the number of clusters tends to be relatively high.

According to [12], incremental clustering constitutes a good choice when the application meets the following criteria. First of all, when the data cannot be easily represented in an n-dimensional space, but it is possible to calculate similarity measures among individuals. Additionally, if the data computation has a high cost and approximate results may be accepted, incremental clustering is faster, allowing us to perform on-line tasks. Finally, if the number of clusters is unknown, incremental clustering methods set the necessary number of clusters depending on different thresholds. These conditions are met in our work since a similarity measure between global-appearance descriptors can be calculated, the method employed does not need to calculate an affinity matrix, easing online operation and the most suitable number of clusters is not known beforehand. Our proposal makes use of omnidirectional images and global-appearance descriptors. Specifically, we used HOG and gist descriptors.

3. Review of Global Appearance Descriptors

In this section, some alternatives to extract global information from panoramic images are summarized. They are known as global-appearance descriptors and they try to keep relevant data with low memory requirements. The visual sensor used in the experiments takes omnidirectional images, for that reason the first step is to transform them into panoramic ones. The starting point is a panoramic image $i(x,y) \in \mathbb{R}^{N_x \times N_y}$ and after using any of the global appearance methods the result is a descriptor $\vec{d} \in \mathbb{R}^{t \times 1}$ where t is the size of the descriptor, as detailed in the next subsections.

Firstly it is necessary to divide the panoramic image in a set of cells. Depending on the shape and number of cells, a different descriptor is obtained. In this work descriptors were built in two different ways. The classic method, described in [48], divides the image in uniformly distributed and non-overlapped horizontal cells. The main idea is that by using panoramic images the descriptor will be invariant to pure rotations of the robot in the ground plane. That is possible due to the fact that the information in the each row is the same and the only change is a horizontal shift in each row. This option was tested in some navigation tasks such as pure visual localization [38], hierarchical localization [3] or topological maps compression [9]. The second method used in this paper consists of defining a set of vertical cells with some overlapping between consecutive cells. To obtain a matching method invariant to the robot orientation, two steps are needed. The descriptor is built by putting together information from the vertical cells, so, the algorithm that compares two descriptors needs, first, to calculate the difference of orientation between both descriptors and shift one of them, by removing its first columns and appending them at the end of the vector. To achieve enough resolution in this step, the descriptor is built with overlapping between consecutive cells. Once the relative orientation is the same, both descriptors can be compared in a straightforward way. Using this additional step that normalizes the orientation, this method also becomes invariant to robot orientation. A comparison between these two methods to build global-appearance descriptors is introduced in [49]. Figure 1 shows how the cells are defined in each of the two methods.

Throughout the paper, the descriptors calculated with horizontal cells are named position descriptors, whereas the descriptors calculated with vertical cells are named orientation descriptors. The objective of using both approaches to build descriptors is twofold. On the one hand, information obtained by descriptors based on pure robot position and by descriptors that are influenced on its orientation are taken into account. On the other hand, this idea could help reduce perceptual aliasing, (i.e., different locations may generate similar visual descriptors). Combining information obtained by horizontal and vertical cells can provide more reliable results as far as image matching is concerned.

These techniques are invariant against changes in the orientation of the robot if it moves in the floor plane and panoramic images are used. To capture them the mobile robot is equipped with an omnidirectional vision system. This system consists of a camera and a hyperbolic mirror mounted in front of the lens. The system is mounted vertically on the mobile robot, as done in several previous works such as [50–53]. The omnidirectional camera captures images with a field of view of 360° around the robot so they offer complete information from the surroundings of the robot from every capture point. Finally, the omnidirectional images might be transformed into panoramic images. The complete experimental setup is presented in Section 5.

Once the methods to divide the image were explained, different approaches to efficiently and robustly describe each region are presented. The Histogram of Oriented Gradient and a descriptor based on gist are the methods used to describe the cells in this work. Table S1 presents the most relevant parameters used in the description process and Table S2 summarizes the parameters that impact the size of the final descriptor.

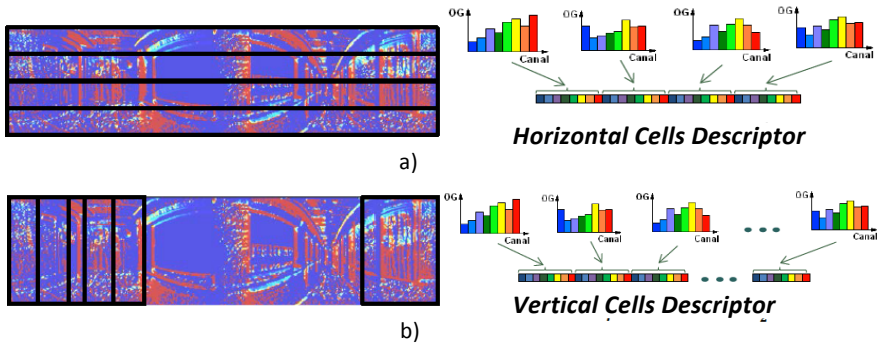


Figure 1. Approaches to build a global-appearance descriptor from a panoramic image: (**a**) with horizontal and (**b**) with overlapped vertical cells.

3.1. Histogram of Oriented Gradient

The Histogram of Oriented Gradient (HOG) was initially used in computer vision to solve object detection tasks. HOG was described by Dalal and Triggs [29], who used it to detect people. This method was later improved in detection and computational cost in the version described in [54]. Afterwards, it was updated by Hofmeister et al. [55], who used a weighted histogram of oriented gradients in small and controlled environments to solve localization from low resolution images. The same authors present a comparison of HOG with other techniques in localization tasks of small robots in reduced environments in [56]. Originally, HOG was built to describe local parts of the scene but it can be redefined to work as a global-appearance descriptor, as in [3], where HOG and other global-appearance descriptors are used to perform hierarchical localization in topological models.

Essentially, it consists of calculating the gradient of the image, obtaining the module and orientation of each pixel. If D_x and D_y represent the derivatives of the image with respect to the x and y axes, respectively, it is possible to calculate the magnitude and orientation of the gradient as:

$$|G| = \sqrt{D_x^2 + D_y^2} \tag{1}$$

$$\theta = \arctan \frac{D_x}{D_y} \tag{2}$$

Afterwards, using a set of cells that covers the whole image it is possible to build the global descriptor using the information of the gradient orientation. To this end, the data is collected in bins, weighting each bin with the module of the gradient of each pixel. Each cell has its own associated histogram and at the end the vector is built concatenating all the histograms. To build the descriptor, the number of bins and cells must be defined. Specifications are shown in Section 6.1.

Classically, this method divided the image into horizontal cells. Authors such as Cebollada et al. [11] or Román et al. [38] used this classical HOG technique to perform localization with a mobile robot. In the present work, as stated before, we will consider both horizontal (with positional intent) and overlapped vertical cells (with orientational intent), as explained at the beginning of the section (Figure 1) . HOG using only horizontal cells is comprehensively described in [48] whereas the second alternative is presented in [49]. Figure 2 shows the method used to build the HOG descriptor, using horizontal cells.

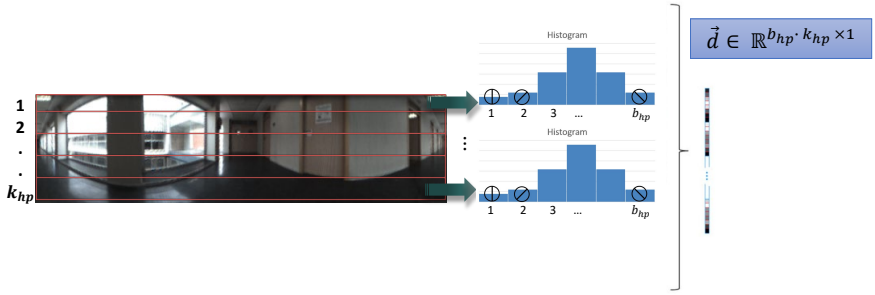


Figure 2. Process to build the HOG descriptor using horizontal cells.

First, regarding the position descriptor, horizontal cells (with the same width as the image) are used, as shown in Figure 1a. Second, in order to obtain the orientation descriptor, vertical cells (with the same height as the image) are used (Figure 1b). Vertical cells are overlapped and separated a distance of $dist_{ho}$ pixels. By shifting the descriptor a rotation of the robot can be simulated.

Once the HOG description is performed, the whole image is reduced to a vector whose size will depend on the number of cells and bins as Figure 1. First, the position descriptor, b_{hp} is the number of bins per histogram and k_{hp} is the number of cells in which the image was divided. The descriptor size is $\vec{d_p} \in \mathbb{R}^{b_{hp} \cdot k_{hp} \times 1}$. Second, in the case of the vertical cells descriptor, the cells are overlapped. This descriptor introduces a new parameter $dist_{ho}$ which refers to distance between consecutive cells. k_{ho} is the number of cells, and it is calculated by $N_y / dist_{ho}$, where N_y is the number of columns of the panoramic image. b_{ho} is still referring to the number of bins per histogram. At the end, HOG with vertical cells reduces a panoramic image into a vector whose size is $\vec{d_o} \in \mathbb{R}^{b_{ho} \cdot k_{ho} \times 1}$. The parameters of the descriptors are summarized in Tables S1 and S2, and the values used in the experiments are specified in Section 6.1.

3.2. Descriptor Based on Gist

The descriptor based on gist was initially introduced in [57] and extended in [58]. This descriptor was further developed in studies such as [33] where it was tested in outdoor environments and as a result, the authors obtain a descriptor whose computational cost is relatively reduced. Some other applications can be found in [59], where a navigation system based on gist is tested; in [60], where gist is calculated in panoramic images and is used to solve localization in urban zones; in [61], where a descriptor based on gist is calculated and dimensionally reduced by using Principal Components Analysis and subsequently used to solve loop closure problems. Finally, Cebollada et al. [11] use such methods to create clustering methods and to perform Visual Place Recognition.

The version of the descriptor used in the present work is built from intensity information, obtained after applying some Gabor filters with different orientations to the image in several resolution levels. To reduce the volume of data each filtered image is divided in a set of cells, and the average intensity of each cell is calculated. As in HOG, classical methods divided the image using horizontal cells [11], whereas new alternatives also tried to use this descriptor using vertical ones [49]. Definitions used in this paper are presented in Table S2 and comprehensively described in [48,49]. Figure 3 shows the process to build the gist descriptor using horizontal cells.

Following that, once the image is filtered with the different masks and scales, the algorithm divides each resulting image into horizontal cells (position descriptor) or into overlapped vertical cells (orientation descriptor), as seen in Figure 1, and the average intensity inside each cell is calculated.

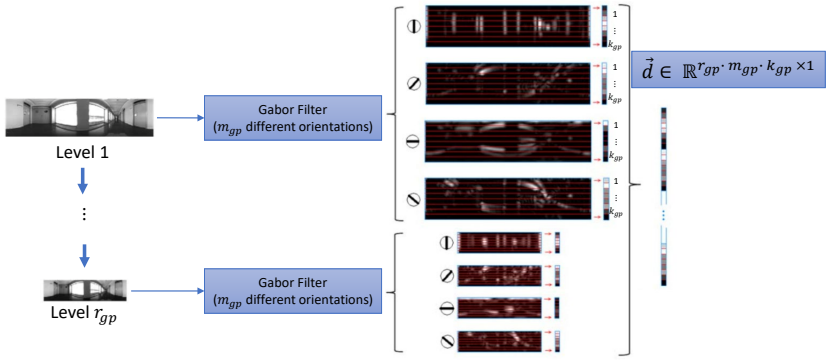


Figure 3. Process to build the gist descriptor using horizontal cells.

In the horizontal-cells descriptor, m_{gp} indicates the number of orientations of Gabor filters, k_{gp} designates the number of cells in which the image was split and r_{gp} indicates the number of different resolution levels used. With these parameters the image can be reduced to a position descriptor whose size is $\vec{d_p} \in \mathbb{R}^{r_{gp} \cdot m_{gp} \cdot k_{gp} \times 1}$. When the cells are vertical, $dist_{go}$ indicates the distance between the beginning of consecutive cells. This parameter is related to the number of vertical cells k_{go} since $k_{go} = N_y / dist_{go}$, where N_y is the number of columns of the panoramic image. m_{go} is the number of orientations of Gabor filters and r_{go} indicates the number of different resolution levels used in orientation descriptor. The orientation descriptor using gist is a vector whose size is $\vec{d_o} \in \mathbb{R}^{r_{go} \cdot m_{go} \cdot k_{go} \times 1}$. The parameters of the descriptors are summarized in Tables S1 and S2, and the values used in the experiments are specified in Section 6.1.

4. Hierarchical Incremental Maps

This section presents the method that we propose to create topological maps incrementally. The starting point is a set of images captured from the same area of the environment, and a node composed of these images. The aim of the process is to build a hierarchical and incremental map, where visually similar zones are detected, compacted and represented as a node. It will be carried out by clustering images with similar features. Broadly speaking, the hierarchical map is built in such a way that when a group of new images do not belong to a previously visited node, a new node is created with them. The hierarchical incremental map is updated and extended every time the robot captures a new image or group of images. A summary of all the parameters needed to follow the process is included in Table S3 and some tunable parameters that can modify and improve the mapping results are shown in Table S4.

When a newly acquired image I_q arrives, firstly, a Node Level Loop Closure is performed with the nodes $N^C = \{N_1, N_2 ..., N_C\}$ currently contained in the map. N^* will represent the set of candidate nodes, so if the node N_i leads to the loop closure, N_i is elected as candidate node and it becomes part of N^*. After retrieving the set of candidate nodes to which the image I_q may belong, secondly, an Image Level Loop Closure is performed with the images that belong to its nodes I^{N^*}.

To carry out these two processes, the position and orientation descriptors are obtained from image I_q. Descriptors should be able to retrieve properly both the candidate nodes and the image that better matches I_q among the images contained in the candidate nodes. At this point, position and orientation descriptors will be only compared with the reference images contained in the nodes which were selected after Node Level Loop Closure $I^{N^*} = \{\cup_{N_i \in N^*} I^{N_i}\}$, where I^{N_i} is the set of images belonging to the node N_i and N^* is the set of nodes selected in the Node Level Loop Closure process. If the Image Level Loop Closure process is successful, a unique image is retrieved as match. If the retrieved image I_i fulfills the Prominence Condition (Section 4.3) and the Centroid Condition Section (4.4), I_q is added to the corresponding node.

Additionally, it is possible that no node is selected in the Node Level Loop Closure process and N^* becomes an empty set ($N^* = \emptyset$). When several consecutive images produce this result ($N^* = \emptyset$) in the Node Level Loop Closure process, a new cluster is created, expanding the hierarchical incremental map. Figure 4 shows a schematic overview of the hierarchical mapping method. The next subsections describe in detail these processes.

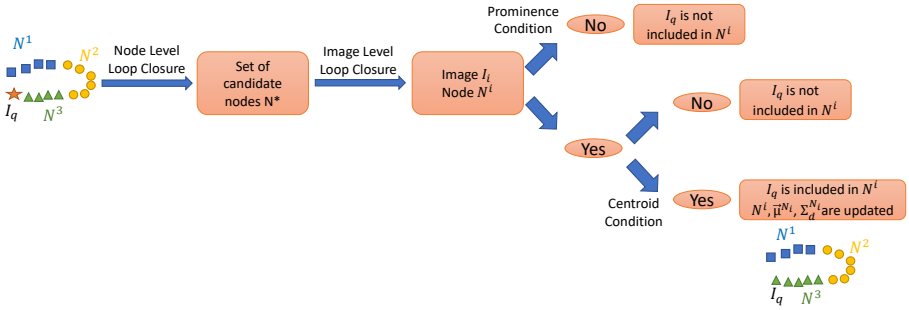


Figure 4. Graphical summary of the proposed method to create hierarchical maps incrementally.

4.1. Node Level Loop Closure

The similarities between each reference node and a new image I_q are evaluated using this method. Nodes are clusters that represent compact zones of the environment, and they contain images with similar features. Each image is described using one of the global appearance descriptors presented in Section 3. Each node N_l is represented through the mean descriptor $\vec{\mu}^{N_l}$ and the covariance matrix $\sum_d^{N_l}$ computed from the descriptors of the images contained in the node.

Similarities between a node and an image are evaluated using the Mahalanobis distance [62]. If $\vec{d}_q$ is the descriptor of the image I_q, the distance $\Delta_{\vec{d}_q}^{N_l}$ between the descriptor $\vec{d}_q$ and the node N_l can be calculated as:

$$\Delta_{\vec{d}_q}^{N_l} = (\vec{d}_q - \vec{\mu}^{N_l})^T (\sum {}_d^{N_l})^{-1} (\vec{d}_q - \vec{\mu}^{N_l}) \quad (3)$$

where i = 1,2,...,C and C is the current number of clusters.

To decide which are the candidate nodes for the closure, the *Mahalanobis distance* has to satisfy the condition of similarity presented in Equation (4), where $\mu_{ns}^{N_l}$ and $\sigma_{ns}^{N_l}$ are the mean and the standard deviation of a Gaussian distribution. When every node is built, 80% of the images are used to model the node, creating with them the mean descriptor $\vec{\mu}^{N_l}$ and the covariance matrix $\sum {}_d^{N_l}$. The other 20% of images are used to build a Gaussian distribution, where $\mu_{ns}^{N_l}$ and $\sigma_{ns}^{N_l}$ represent mean and standard deviation of the distances between each of these 20% of images to the mean descriptor of the node. Also in Equation (4), x is a parameter that must be tuned, and whose value must depend on the number of clusters. The lower x, the more restrictive the condition to create a new node. Values of x used in this work vs. number of clusters can be seen in Figure 5. As shown in this figure, x is less restrictive for a low number of clusters, but it is necessary to limit it from a specific number of clusters. This limit is established depending a parameter (Ω) which has to be tuned (Table S4). x remains constant at {1.7, 1.85, 2, 2.15, 2.3} when $C \geq 9, C \geq 8, C \geq 7, C \geq 6$ or $C \geq 5$ respectively .

$$\left| \Delta_{\vec{d}_q}^{N_l} - \mu_{ns}^{N_l} \right| \leq \left| x \sigma_{ns}^{N_l} \right| \quad (4)$$

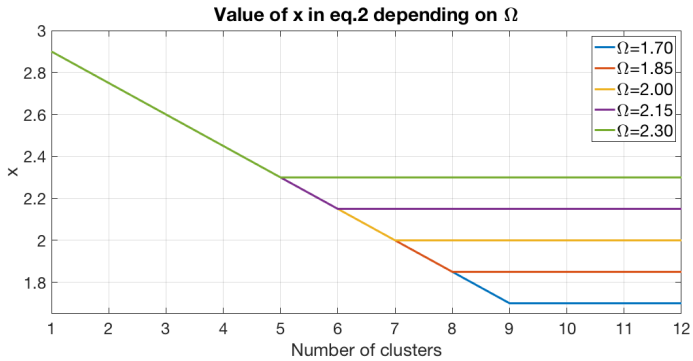


Figure 5. Values of x in the node level loop closure condition versus number of clusters in Equation (4).

At the end of the Node Level Loop Closure, all the nodes that satisfy Equation (4) are considered candidate nodes and introduced in the set N^*. After that, the Image Level Loop Closure is activated to try to retrieve the specific image from these nodes that closes the loop. In some occasions, the Node Level Loop Closure may not find any candidate node that closes the loop. In that case, it outputs an empty set of nodes, and it is considered that the new image does not close the loop with any of the existing nodes so far.

4.2. Image Level Loop Closure. Position and Orientation Descriptors

This algorithm activates if the Node Level Loop Closure is successful and outputs a non-empty set N^* of candidate nodes. In this case, it is necessary to determine which specific image or images close the loop, being the most similar to I_q. Given the set of candidate nodes N^*, all their corresponding images I^{N^*} are evaluated to find the image I_i which is the most similar image to I_q. The problem is solved using two different similarity metrics, described in Section 3. These similarity metrics are computed between the descriptor of the image I_q and every of the descriptors of the I^{N^*} candidate images. Using both metrics, the comparison combines position and orientation information.

First, position information is obtained comparing the position descriptor of the image I_q with the position descriptors of the images I^{N^*} using *Euclidean distance*. Second, the orientation information is used. In this case the method estimates the relative shift between each candidate image I_i and I_q. Then the panoramic image I_i and its associated descriptor are shifted, in such a way that after this shift, both images have the same relative orientation. Then the distance between the orientation descriptors is calculated using *Euclidean distance* (Equation (5)), where $\vec{d}_q$ is the descriptor obtained from the new image I_q and $\vec{d}_k$ is the descriptor of each of the images I_k contained in the candidate nodes N^* after running the Node Level Loop Closure process.

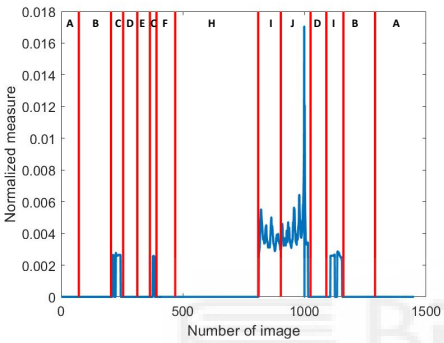
$$dist_{eucl}{}^{\vec{d}_k}_{\vec{d}_q} = \sqrt{\sum_{j=1}^{t} ((\vec{d}_q(j) - \vec{d}_k(j))^2} \tag{5}$$

This process is repeated for all the images I_k contained in the set of candidate nodes N^*. After that, the method calculates the inverse of these distance measures, so the images with small distances (images that are more similar) will have higher similarity measure. Then, the result is normalized in such a way that the sum of the similarities is equal to 1. Next, the position and orientation similarity measures are multiplied to obtain the final similarity measure that combines both kinds of information. The image I_i with the associated higher result is then retrieved by the Image Level Loop Closure (and its node as the most similar node to I_q. It is possible to see an example of these measures in Figure 6, where Figure 6a shows the similarity measures between the image I_q and the images in the candidate nodes when using position descriptors; Figure 6b shows the same similarity comparison but using orientation descriptors and Figure 6c shows the final similarity measures obtained by multiplying

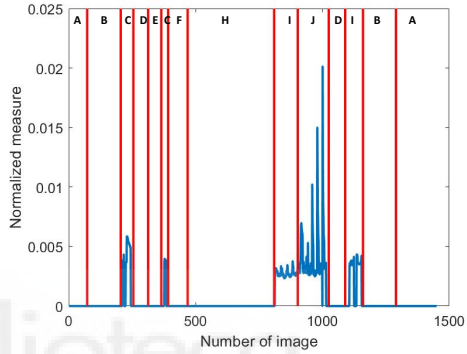
position and orientation measures. These results (final similarity measure) are the data used to retrieve both the image and the node that close the loop with I_q. Only the images I_k that belong to the candidate nodes N^* have a value in these figures, the other images are considered to have null measure.

$$sim^{\vec{d}_k}_{\vec{d}_q} = \frac{1}{dist_{eucl}{}^{\vec{d}_k}_{\vec{d}_q}} \tag{6}$$

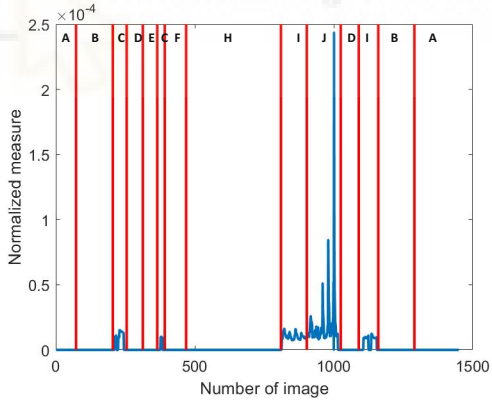
$$i = \underset{k}{\text{argmax}}(sim^{\vec{d}_k}_{\vec{d}_q}) \tag{7}$$



(a) Similarity between position descriptor of I_q and position descriptors of the images in the candidate nodes.



(b) Similarity between orientation descriptor of I_q and orientation descriptors of the images in the candidate nodes.



(c) Final similarity between descriptor of I_q and descriptors of the images in the candidate nodes.

Figure 6. Node Level Loop Closure process example. The node labels are arranged along the top of each subfigure. In this example, the Node Level Loop Closure process has previous retrieved nodes C, I and J.

If no node is selected as candidate node ($N^* = \emptyset$) the Image Level Loop Closure process cannot be run. Therefore, the image I_q is not assigned to any node, and it remains unclassified, waiting for information from the next images (those which are subsequently captured by the robot). When a set of consecutive images are left unclassified, a new node has to be created with them and the node

representatives are recalculated (the node representative is the mean descriptor of the images contained in the node).

4.3. Prominence Condition

As explained in Section 4.1, the Node Level Loop Closure process selects N^* possible nodes. Among them, the Image Level Loop Closure process finds the most similar image (I_i) which closes the loop. The node to which I_i belongs, is the finally selected node. Prior to selecting this node N_i, the image detected as the most similar to I_i should meet a prominence condition.

The prominence measures how much the peak of the similarity curve (Figure 6c) stands out due to its intrinsic height and its location relative to other peaks. This condition is taken into account because not only a high peak should occur to close the loop, but also that peak should be very distinct compared to its neighbours. The image selected from the Image Level Loop Closure should fulfil the condition presented in Equation (8). In this equation, P_{I_i} is the prominence value of the candidate image, $\mu(P_{I_{k^*}})$ is the mean prominence of the candidate images contained in N^*. γ is a threshold, which is empirically tuned from the experiments as $\gamma = 5$.

$$P_{I_i} \geq \gamma * \mu(P_{I_{k^*}}) \tag{8}$$

4.4. Centroid Condition

As detailed in the previous subsections, the node Level Loop Closure selects first N^* candidate nodes. Among them, the Image Level Loop Closure finds the most similar image which closes the loop and helps select the final node. In order for a selected node to be accepted as the final node, it must fulfil the condition of Equation (9) below. This condition evaluates how the position of the node representative shifts when the new candidate image is added to the selected node. In the condition, the distance between the node representative before (μ^{N_i}) and node representative after adding the descriptor of the new image ($\mu^{N_i \cup \{\vec{d}_q\}}$) has to be lower or equal to the maximum influence on the node representative effected by the images already contained in the node.

$$\left| \vec{\mu}^{N_i} - \vec{\mu}^{N_i \cup \{\vec{d}_q\}} \right| \leq \max_j \left(\left| \vec{\mu}^{N_i - \{\vec{d}_j\}} - \vec{\mu}^{N_i} \right| \right) \tag{9}$$

where $\vec{d}_j$ denote all descriptors of the images contained in the node N_i.

The process presented in Sections 4.1–4.4 is summarized in the pseudocode Algorithm 1. Starting from the current set of clusters in the map $N^C = \{N_1, N_2 ..., N_C\}$, their associated data (N_l,$\vec{\mu}^{N_l}$ *and* $\sum_d^{N_l}$ for $l = 1, 2, ..., C$) and the descriptors of each previous image $\vec{d}_k$, it is possible to determine if a new image I_q, whose descriptor is $\vec{d}_q$, has to be assigned to any previous node or not.

Algorithm 1 Pseudocode for Node and Image Level Loop Closure, applying Prominence and Centroid Conditions.

NodeLevelLoopClosure ($\vec{d}_q$)
$N^C = \{N_1, N_2..., N_C\}$ initial set of clusters
$\vec{\mu}^{N_l}$, $\sum_d^{N_l}$ mean descriptor and covariance matrix of cluster N_l
for `l=1 to number of clusters C` **do**
    $\Delta_{\vec{d}_q}^{N_l} = (\vec{d}_q - \vec{\mu}^{N_l})^T (\sum_d^{N_l})^{-1} (\vec{d}_q - \vec{\mu}^{N_l})$ (Equation (3))
    if $\left|\Delta_{\vec{d}_q}^{N_l} - \mu_{ns}^{N_l}\right| \leq \left|x\sigma_{ns}^{N_l}\right|$ **then** (Equation (4))
        $N^* \leftarrow N_l$;
    end if
end for
end NodeLevelLoopClosure
ImageLevelLoopClosure ($\vec{d}_q$, $\vec{d}_k$, N^*)
for `k= 1 to images in` N^* **do**
    $dist_{eucl}{}_{\vec{d}_q}^{\vec{d}_k} = \sqrt{\sum_{j=1}^{t} ((\vec{d}_q(j) - \vec{d}_k(j))^2}$ (Equation (5));
    find the most similar image I_i using Equation (7);
    if I_i meets $P_{I_i} \geq \gamma * \mu(P_{I_{k^*}})$ **then** (Equation (8), Prominence Condition)
        if I_i meets $\left|\vec{\mu}^{N_i} - \vec{\mu}^{N_i \cup \{\vec{d}_q\}}\right| \leq \max_j(\left|\vec{\mu}^{N_i - \{\vec{d}_j\}} - \vec{\mu}^{N_i}\right|)$ **then** (Equation (9) Centroid Condition)
            I_i and its node N_i close the loop;
            I_q is included in N_i;
            N_i, μ^{N_i} and $\sum_d^{N_i}$ are updated;
        else
            I_i does not meet Equation (9);
            I_q is not included in N_i;
        end if
    else
        I_i does not meet Equation (8);
        I_q is not included in N_i;
    end if
end for
if $N^* = \emptyset$ **then**
    I_q is not assigned to any node;
    The existing nodes are not updated and I_q will be evaluated in subsequent steps;
end if
end ImageLevelLoopClosure

4.5. New Node Creation

The methods presented so far detect the node to which the new image I_q belongs. However, as explained before, the new image may not be assigned to any previous node. That may occur either because the new image is quite different to the existing node representatives or because it does not fulfill either the prominence condition (Section 4.3) or the centroid condition (Section 4.4).

Once a considerable number of images consecutively captured are considered not to belong to any cluster, a new cluster N_q is created. The new cluster is modelled with its mean descriptor $\vec{\mu}^{N_q}$ and covariance matrix $\sum_d^{N_q}$. In order to obtain a new cluster which is significantly different to the ones already existing in the map, the average distance between the representative of the new node and the representative of the other nodes (using Euclidean distance) has to be higher or equal to the average distance among the representatives that already exist. If the new cluster does not fulfill this

condition, this new cluster N_q cannot be elected in further Node Level Loop Closure processes. In this way, new images continue to be evaluated without considering this cluster but if they are still not assigned to any node but they are similar enough to the cluster N_q, they are included to that cluster until the distances from the representative of N_q to other representatives is higher or equal to the average distance among the representatives of the other clusters in the map. This condition tries to guarantee that clusters are different enough among them and that they do not spread excessively along the space of features.

4.6. Node Merging

According to the previous processes, the number of clusters constantly increases. Therefore the resulting number of clusters is sometimes higher than necessary and images that represent adjacent and/or visually similar zones may be included in a unique cluster. The proposed method has the possibility to decrease the number of clusters by merging similar nodes when necessary. That is possible because the method detects when two clusters are similar and the map would be more consistent if they are merged. A new condition is introduced and a newly created node N_q has to exceed a specific dissimilarity threshold to the other clusters. If an existing cluster and a new one are similar, they are merged in a unique cluster. To solve this problem the Mahalanobis distance is used as in Node Level Loop Closure process. Equation (10) shows this condition in the node merging process

$$\Delta_{N_q}^{N_o} = (\vec{\mu}^{N_q} - \vec{\mu}^{N_o})^T (\sum {}_{d}^{N_o})^{-1} (\vec{\mu}^{N_q} - \vec{\mu}^{N_o}) \tag{10}$$

where l = 1,2,. . . ,C and C is the number of clusters before creating N_q. To detect if the node N_q should merge with another node N_l, $\Delta_{N_q}^{N_o}$ is evaluated. This distance has to satisfy the condition of similarity presented in Equation (11), where $\mu_{ns}^{N_o}$ and $\sigma_{ns}^{N_o}$ are the mean and the standard deviation of the Gaussian distribution calculated with the data of node N_o, calculated as detailed in Section 4.1. y is a parameter which has to be empirically tuned (Table S4). The higher it is, the less restrictive the condition is, so the algorithm is more prone to merge similar clusters. Experiments show that this threshold should take a value between 1 and 2. If $y = 1$ the condition is very restrictive so less mergers are done and more clusters will be in the final map, but when $y = 2$ the condition is less restrictive, so more mergers are done and less clusters will be at the end. Experiments to test the values of y are carried out in Section 6.

$$\left|\Delta_{N_q}^{N_o} - \mu_{ns}^{N_o}\right| \leq \left|y\sigma_{ns}^{N_o}\right| \tag{11}$$

5. Image Sets for Experiments

The proposed algorithms are tested with several sets of images captured under real operating conditions. The datasets used are the INNOVA dataset [22] captured by ourselves and the COLD database [53], a third party publicly available dataset which provides robot trajectories in some buildings of the Freiburg and Saarbrücken universities. These three trajectories provide a good choice to perform hierarchical incremental mapping because they represent real environments that experience the typical phenomena that can occur during real operation such as noise, occlusions of the images, changing lighting conditions, movement of people or even some objects or pieces of furniture etc. For that reason these sets of images constitute a challenging scenario to test the robustness of the incremental mapping algorithms.

The different image sequences are recorded by a mobile robot, which is equipped with an omnidirectional camera. The catadioptric vision system is made using a hyperbolic mirror mounted in front of the camera on a portable bracket. The program receives the omnidirectional images and it transforms them into panoramic ones and starts the mapping process, updating the map every time that a new image I_q arrives according to Section 4. Additionally, the robot is equipped with wheel encoders and a laser range scan, which are used to obtain the ground truth, for comparative purposes. However, the proposed method does not use these data and it carries out the incremental mapping

with pure visual information. Figure 7 shows the robot and the camera used to capture the INNOVA dataset. Complete information about the equipment used to capture the COLD dataset can be found in reference [53]. Figure 8 shows some sample images from all the datasets.

Figure 7. Mobile robot and its vision system in INNOVA dataset [22].

(**a**) Image from INNOVA

(**b**) Image from Saarbrücken

(**c**) Image from Freiburg

Figure 8. Panoramic images from each of the datasets.

Table 1 shows some specifications about the 3 trajectories and the number of images taken in each trajectory. Each route contains some loops so they permit testing the Node Level and Image Level Loop Closure processes. Finally, we consider essential to say that these environments are a challenging choice because across the route many walls are made of glass and there are lots of windows so the outdoor weather and lighting conditions could have a negative impact upon the mapping task.

Table 1. Distance covered [m] and number of images used for each of the datasets.

Trajectory Dataset	Number of Images	Distance Covered
Route 1 INNOVA	1450	176.26 m
Route 2 Saarbrücken	1021	56.64 m
Route 3 Freiburg	2778	102.68 m

6. Experiments

In this section, the methods proposed for hierarchical incremental map creation are tested using the image sets introduced in the previous section. Firstly some images are taken to create the first cluster in a supervised method. Once the first cluster is created, the process can start in order to create new clusters, incrementally updating the map every time a new image I_q arrives.

6.1. Parameters to Describe the Images

As explained in Section 3, using global-appearance descriptors, each panoramic image is reduced to a vector $\vec{d} \in \mathbb{R}^{l\times 1}$ whose size depends on the parameters used in this description process. These parameters are outlined in Table S2. In this work, we use $b_{hp} = k_{hp} = 16$ with the HOG position descriptor. In this case, the position vector size is $\vec{d}_h \in \mathbb{R}^{256\times 1}$. For the orientation descriptor $b_{ho} = 16$, $k_{ho} = 256$ and $dist_{ho} = 2$, so the panoramic image is reduced to a vector whose size is $\vec{d}_v \in \mathbb{R}^{4096\times 1}$. In the case of gist, the parameters were kept constant in $m_{gp} = k_{gp} = 16$ and $r_{gp} = 2$ for the position descriptor. Using these parameters the descriptor size is $\vec{d}_p \in \mathbb{R}^{512\times 1}$. In the case of the gist orientation descriptor, it is calculated with $m_{go} = 16$, $k_{go} = 256$, $dist_{go} = 2$ and $r_{go} = 1$ and each image is transformed to a vector $\vec{d}_o \in \mathbb{R}^{4096\times 1}$.

6.2. Parameters to Perform the Loop Closure Processes

When a new image arrives, the first step is to know if it belongs to an existing cluster. As detailed in Section 4, to perform the Node Level Loop Closure, the *Mahalanobis distance* is used, (Equation (3)). After that, it is possible to detect if the image descriptor $\vec{d}_q$ may belong to a specific node N_i, storing all the candidate solutions in the set N^*.

Once the candidate nodes are selected, their images are compared with $\vec{d}_q$ using *Euclidean distance*, Equation (5), where d^{N_k} are all the candidate descriptors from the Node Level Loop Closure. The Image Level Loop Closure detects the most similar image among them, named I_i. After detecting I_i, the image has to fulfill the Prominence and Centroid Conditions (Sections 4.3 and 4.4). Furthermore, the process evaluates if two nodes are similar enough to be merged (Section 4.6).

As described in Section 4, these equations depend on different parameters that have an influence on the results. Equation (4) depends on x. It establishes how restrictive the process is to close the loop in the node level. The lower x is, the easier to close the loop. x must depend on the number of clusters (C) and it is tuned as $x = 3.05 - 0.15 * C$; with limits depending on Ω, Figure 5 shows graphically these values. Another parameter is y, which is used in Equation (11) and it influences the node merging; the lower y, the easier to merge similar nodes. Finally, the parameter γ appears in Equation (8). Once I_i is retrieved, this equation evaluates the prominence of the result to ensure that the retrieved image has a peak on the similarity curve higher enough compared to its the neighbours. The higher γ is, the more prominent the peak must be. During the experiments γ is constant, $\gamma = 5$. These parameters are summarized in Table S4.

6.3. Evaluation

The relative performance of a clustering framework can be evaluated by means of the silhouette. Silhouette evaluates the compactness of each cluster, i.e., the degree of similarity between each descriptor and the other descriptors of the same cluster, comparing it with the descriptors that belong

to the other clusters. The average silhouette (S) of all the entities (images descriptors) is calculated with Equation (12). The higher S is, the more similar each descriptor is to the descriptors in its own cluster and the more different to the descriptors in the other clusters. The maximum value of the silhouette is 1, indicating that the resulting clusters contain well-separated images; the descriptors in each cluster are very similar among them and different to descriptors in the other clusters. By contrast, the minimum value is -1, which means that the resulting clusters do not separate correctly the information. In this work, the silhouette is used to evaluate the compactness of the clusters. Therefore, instead of using the similarity in the feature space, world coordinates are used.

$$S = \frac{\sum_{i=1}^{C} s_i}{C} \tag{12}$$

In Equation (12), C is number of clusters and s_i the average silhouette of the descriptors contained in the cluster i. The silhouette of each descriptor $\vec{d_j}$ is calculated using Equation (13) .

$$s_j = \frac{b_j - a_j}{max(a_j, b_j)} \tag{13}$$

where a_j is the average distance between the capture point of $\vec{d_j}$ and the capture points of the other descriptors in the same cluster and b_j is the average distance between the capture point of $\vec{d_j}$ and the capture points of the other descriptors in the different clusters. The information about the capture points is known because the ground truth of the data set is available, so this information is used to quantify the performance of the mapping algorithm. Notwithstanding, it is worth highlighting that the mapping task is carried out with pure visual information.

In addition, the number of clusters obtained after the process will also be shown in order to evaluate how the parameters have an influence upon this number.

6.4. Results

6.4.1. Influence of the Parameters on the Performance of the Algorithm

Figure 9 shows the results obtained with the HOG descriptor, and Figure 10 with gist for (a) INNOVA, (b) Saarbrücken and (c) Freiburg datasets. In both figures, the first row shows the number of clusters obtained once all the images of each trajectory were processed by the proposed algorithm. The second row presents the average silhouette of the descriptors. These figures show the influence of the parameters y and Ω, which are introduced in Table S4.

First, considering parameter y, as expected, the higher y is the fewer clusters will be in the final map. The effect of this parameter is less significant when gist is used, as shown Figure 10. Second, Ω limits x, the parameter used in Equation (4). If this value is low, more nodes can be selected on the Node Level Loop closure so it can be more difficult to obtain a high number of consecutive images without being in a cluster so the effect of creating new nodes decreases. However, if this value is too high it may lead to some images in a row not being assigned to any node when no new cluster should be created in the Node Level Loop Closure process. As Figures 9 and 10 show, when HOG is used, the parameter y has more influence on the number of clusters than Ω does, yet with the gist descriptor, the latter has greater influence.

The second row of each subfigure in Figures 9 and 10 shows the average silhouette after the mapping process. In general, medium values of Ω offer a higher silhouette but the effect of y is more variable. To better study the results, Table 2 shows the maximum values of silhouette and the number of clusters obtained with the different datasets. It also shows the values of the parameters that lead to that maximum silhouette. It shows how values of Ω around 1.85 offer the best results although when using gist descriptor good silhouettes can be also obtained with higher Ω values. Meanwhile, the results are not substantially dependent on y. As it is possible to observe, better results are obtained using the HOG descriptor.

This behaviour was observed among different experiments and situations, which makes us conclude that HOG is a more suitable option to describe the images with the objective of creating a map incrementally. In addition, the silhouette obtained with the Freiburg dataset is substantially lower than the silhouette obtained with the two other datasets. The reason can be twofold. On the one hand, the route contains a large number of images, and it has some spaces which are visually similar, which challenges the mapping algorithm. On the other hand, the Freiburg environment contains several glass walls, and large and numerous windows, which produce saturations in the images and mixing between the information of adjacent rooms. These phenomena also have a negative impact upon the performance of the method.

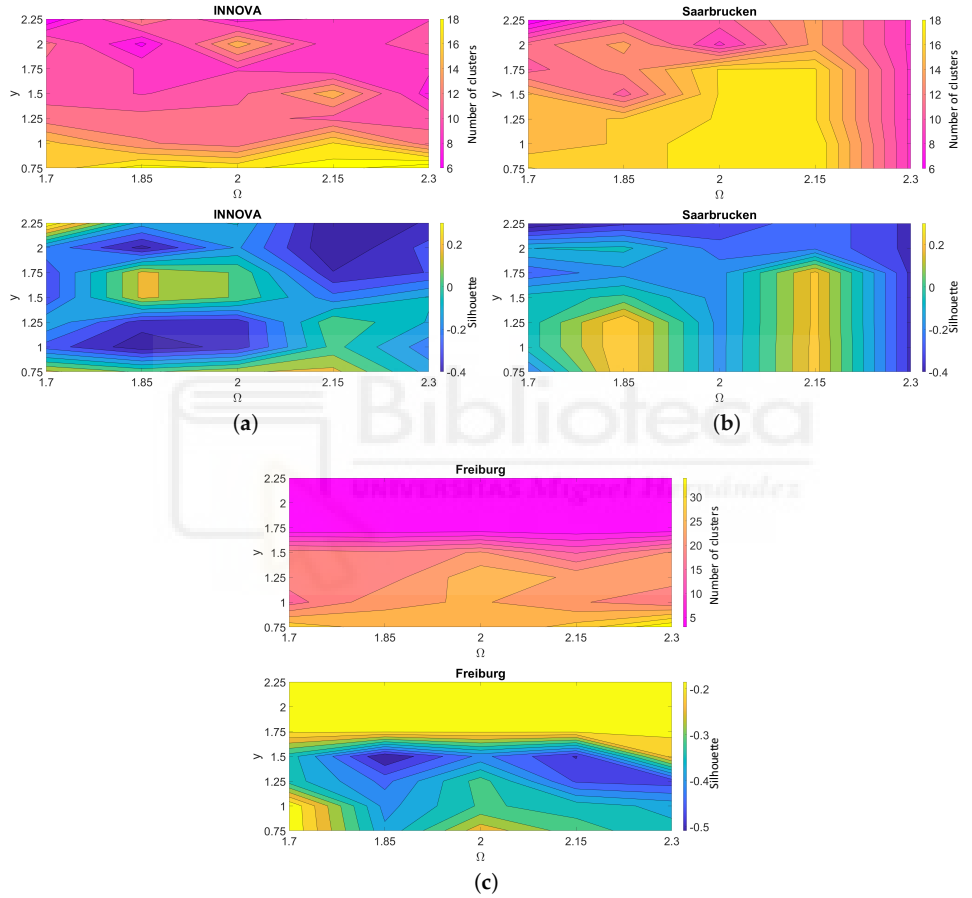


Figure 9. Results obtained with the HOG descriptor for different values of y and Ω for the datasets (**a**) INNOVA, (**b**) Saarbrücken and (**c**) Freiburg. The first row of each subfigure shows the final number of clusters (colour mapped) and the second one the average silhouette (colour mapped) after the proposed incremental hierarchical mapping process.

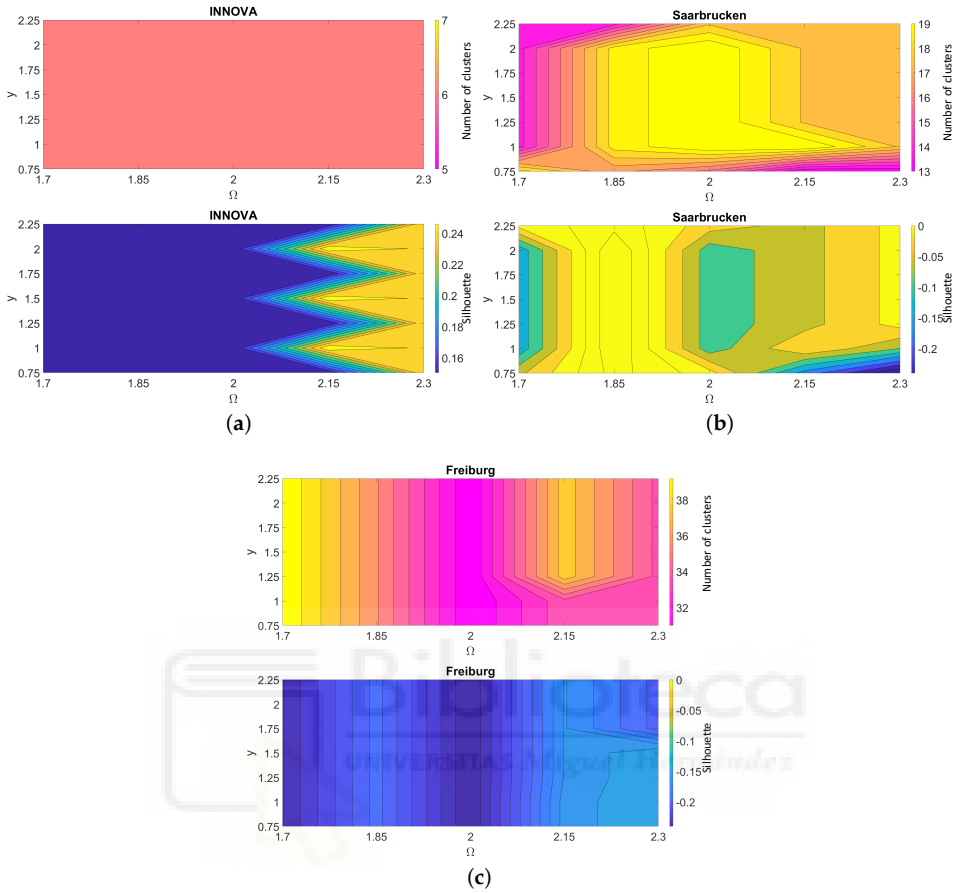


Figure 10. Results obtained with the **gist** descriptor for different values of y and Ω for the datasets (**a**) INNOVA, (**b**) Saarbrücken and (**c**) Freiburg. The first row of each subfigure shows the final **number of clusters** (colour mapped) and the second one the average **silhouette** (colour mapped) after the proposed incremental hierarchical mapping process.

Table 2. Maximum silhouette obtained per configuration, showing also the number of clusters and the configuration of the parameters.

HOG			
Dataset	**Silhouette**	**Number of Clusters**	**Parameters Values**
Route 1 INNOVA	0.3973	6	$y = 2.25, \Omega = 1.7$
Route 2 Saarbrücken	0.2756	16	$y = 1.25, \Omega = 1.85$
Route 3 Freiburg	-0.1526	30	$y = 0.75, \Omega = 1.7$
Gist			
Dataset	**Silhouette**	**Number of Clusters**	**Parameters Values**
Route 1 INNOVA	0.2556	6	$y = 1.5, \Omega = 2.15$
Route 2 Saarbrücken	0.1262	19	$y = 1.5, \Omega = 1.85$
Route 3 Freiburg	-0.1449	34	$y = 1.5, \Omega = 2.3$

6.4.2. Batch Spectral Clustering Results

For comparative purposes, we consider a batch spectral clustering algorithm [11] as benchmark. It is worth highlighting that this batch spectral clustering algorithm has complete information about all the descriptors from the beginning of the process, and can calculate all the mutual similitudes to perform the clustering process. Therefore, it constitutes a powerful benchmark to compare the relative performance of our proposal. To use this batch technique, the number of clusters has to be set initially and all the images must be available, for this reason it is not a good option to create maps incrementally, as the robot explores new areas. Figures 11 and 12 show the results after using a batch spectral clustering method and either HOG or gist descriptors respectively. These figures represent average silhouette versus number of clusters.

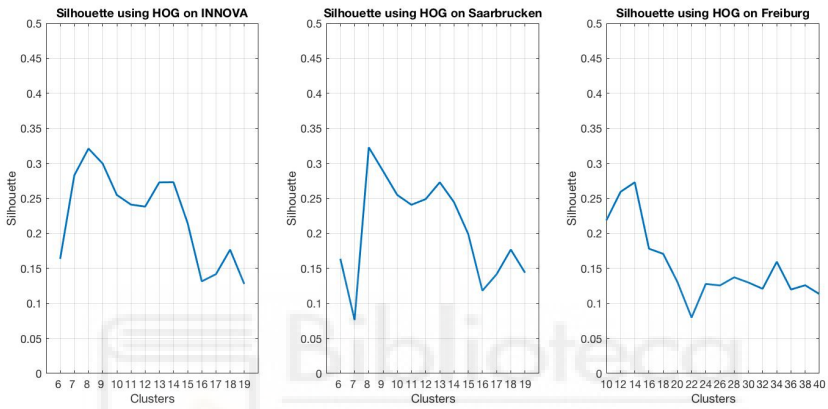


Figure 11. Average silhouette after batch spectral clustering process using HOG versus number of clusters.

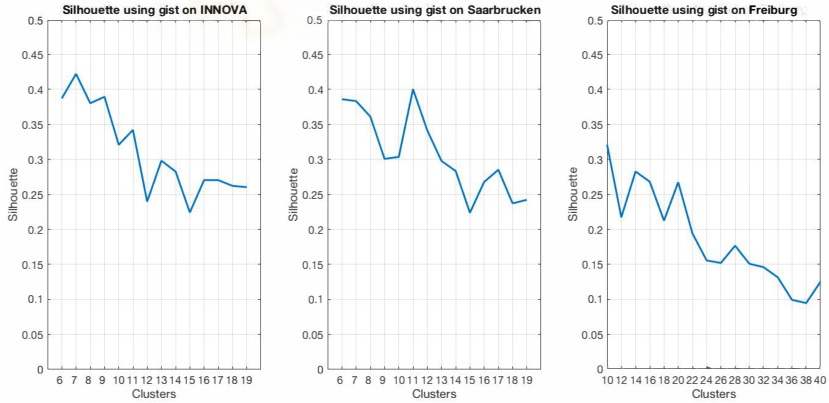


Figure 12. Average silhouette after batch spectral clustering process using gist versus number of clusters.

Figure 11 shows the average silhouette when the HOG descriptor is used. This figure shows that the average silhouette decreases when the number of clusters continuously increases. The results obtained with the INNOVA dataset show that the silhouette is between 0.1 and 0.3; in particular if we observe the result with 6 clusters (which lead to the maximum silhouette with the proposed method) silhouette is 0.13 while with the proposed method is 0.3973. In the case of the Saarbrücken dataset, while the silhouette obtained with batch spectral clustering is between 0.1 and 0.3, the results with proposed incremental method are between -0.4 and 0.3. Observing the result with 16 clusters

(which led to the best silhouette with the proposed method) we note this is about 0.15 as opposed to 0.2756 with the proposed algorithm. Finally, regarding the Freiburg dataset, while the results using batch spectral clustering are between 0.1 and 0.25, results using the incremental method are among −0.5 and −0.15. The maximum value of silhouette of the proposed method is equal to −0.1526 with 30 clusters, while the batch clustering method provides a silhouette of 0.13 with the same number of clusters.

Additionally, Figure 12 show the silhouette when the gist descriptor is used. In this case, the results are less similar to the results obtained with the proposed method. Again, the higher the number of clusters is, the lower the silhouette values are. The best silhouette obtained with gist and the INNOVA dataset is 0.2556 with 6 nodes, the batch clustering method leads to a silhouette equal to 0.38 with the same number of nodes. With the Saarbrücken dataset, silhouette results with a high number of clusters are around 0.25 and 0.4 with the batch method and 0.1262 (19 clusters) with the proposed method. Finally, the results output by the batch clustering method on Freiburg show a silhouette between 0.1 and 0.25 but using the proposed incremental clustering process the maximum silhouette is −0.1449, obtained with 34 clusters. With 34 clusters the silhouette obtained with batch clustering is 0.15. The results of this comparative evaluation between the proposed method and the benchmark algorithm are summarized in Table 3. This table includes the maximum silhouette obtained with the proposed method, and the silhouette obtained with the batch spectral clustering with the same number of clusters. It is necessary to highlight the fact that the proposed method provides better results with HOG in the Innova and Saarbrücken datasets. It is especially relevant, considering that the proposed method permits building the map as the robot explores the environment (i.e., it works with incomplete visual information) but the batch clustering needs to have all the images captured before running the algorithm. The results with the Freiburg dataset are less conclusive, due to the features of this dataset, commented previously. HOG stands out again as an efficient image description method with incremental mapping purposes.

Table 3. Comparison between the maximum silhouette obtained with the proposed method and the silhouette obtained with a batch spectral clustering with the same number of clusters.

HOG			
Dataset	**Number of Clusters**	**Proposed Method**	**Batch Spectral Clustering**
Route 1 INNOVA	6	0.3973	0.13
Route 2 Saarbrücken	16	0.2756	0.15
Route 3 Freiburg	30	−0.1526	0.13
Gist			
Dataset	**Number of Clusters**	**Proposed Method**	**Batch Spectral Clustering**
Route 1 INNOVA	6	0.2556	0.38
Route 2 Saarbrücken	19	0.1262	0.25
Route 3 Freiburg	34	−0.1449	0.15

6.4.3. Bird's Eye View of the Capture Points

Figures 13–15 show a bird's eye view of the capture points of the three datasets in different points of the proposed incremental clustering process. These capture points are shown with different shapes and colors, depending on the cluster they belong to. Figures 13j, 14g and 15j show the result after completing all the proposed incremental mapping method, and the subfigures show how the clusters are being created while the map is updated. Observing the pairs of subfigures: Figure 13c,d,f,g or Figure 15e,f it is possible to see how node merging (Section 4.6) works. Initially, there are several clusters and after including some new images and creating new clusters or making a specific cluster larger, it is possible to obtain similar clusters, so the merging process is launched and it results in a lower number of clusters. Finally, Figure 14b,e shows how transition between rooms resulted in an increased number of clusters. After 399 images, the process has detected 7 clusters but from moment

Figure 14e to the end of the process the robot has moved across a long corridor and only changes the room once, for that reason only 2 clusters are created on that process. Others characteristics such as loop closure can be observed in the figures.

(**a**) 240 images and 3 clusters.

(**b**) 286 images and 5 clusters.

(**c**) 344 images and 5 clusters.

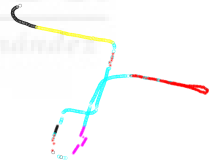
(**d**) 417 images and 4 clusters.

(**e**) 537 images and 8 clusters.

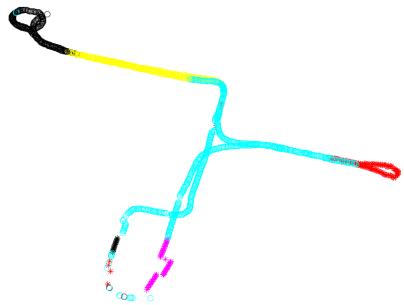
(**f**) 640 images and 10 clusters.

(**g**) 818 images and 11 clusters.

(**h**) 926 images and 12 clusters.



(**i**) 985 images and 6 clusters.



(**j**) 1450 images and 6 clusters.

Figure 13. Maps obtained during the process using HOG $y = 2.25$, $\Omega = 1.7$ with INNOVA dataset. The subfigures show different steps of the process. In the end, the process detects 6 nodes and the final average silhouette is 0.3973. The pairs of **c**,**d**,**f**,**g** show how the node merging process works.

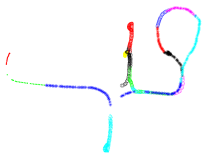
(**a**) 268 images and 3 clusters.

(**b**) 399 images and 7 clusters.

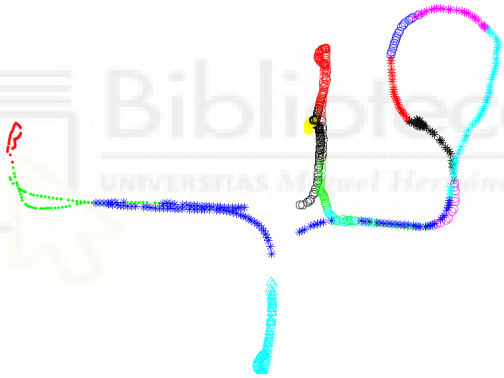
(**c**) 477 images and 9 clusters.

(**d**) 629 images and 13 clusters.

(**e**) 785 images and 14 clusters.



(**f**) 886 images and 16 clusters.



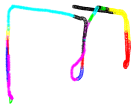
(**g**) 1021images and 16 clusters.

Figure 14. Maps obtained during the process using HOG $y = 1.25$, $\Omega = 1.85$ with the Saarbrücken dataset. The subfigures show different steps of the process. In the end, the process detects 16 nodes and the final average silhouette is 0.2756.

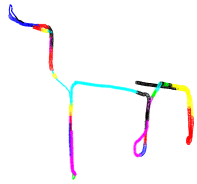
(**a**) 322 images and 3 clusters.

(**b**) 587 images and 9 clusters.

(**c**) 929 images and 13 clusters.

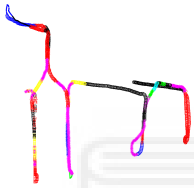


(**d**) 1118 images and 15 clusters.

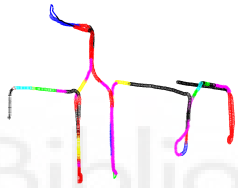


(**e**) 1581 images and 22 clusters.

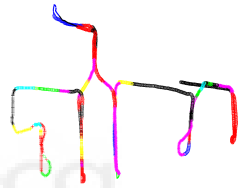
(**f**) 1714 images and 19 clusters.



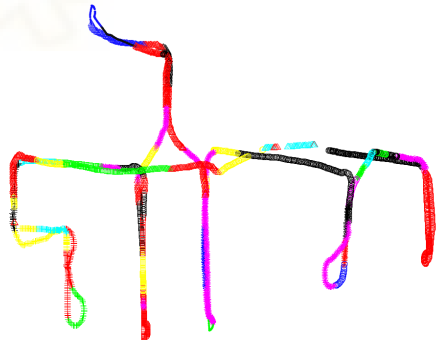
(**g**) 1957 images and 21 clusters.



(**h**) 2102 images and 24 clusters.



(**i**) 2347 images and 28 clusters.



(**j**) 2778 images and 30 clusters.

Figure 15. Maps obtained during the process using HOG $y = 0.75$, $\Omega = 1.7$ with the Freiburg route. The subfigures show different steps of the process. In the end, the process detects 30 nodes and the final average silhouette is -0.1526. The pair of **e**,**f** shows how the node merging process works.

7. Conclusions

This work presented a method to create hierarchical topological maps incrementally, updating the map every time a new image arrives. The framework is based on the development of an incremental clustering algorithm, presented throughout the paper. The experiments were made in real indoor

environments where the robot navigates under real operation conditions, including illumination variations and changes introduced by human activity. The robot is equipped with an omnidirectional vision system, and the only information used to build the hierarchical map are the images captured by this system. To describe the images two different global-appearance methods were considered: HOG and gist. The experimental section showed the performance of the proposed algorithm and the effect of the most relevant parameters on the final result. Also, a comparative evaluation with a batch spectral clustering algorithm is performed.

The relative accuracy of the method is studied by means of the average silhouette, calculated considering as entities the capture points of every image (ground truth). First, HOG proved to output better results, such that the average silhouette obtained with the proposed incremental method is similar to the silhouette output by the batch spectral clustering method despite the fact that the proposed algorithm only has partial information at each step. In fact, if the parameters are tuned properly if can offer better results than the batch spectral clustering. The results show that in the case of HOG, it is especially important tuning correctly the parameter y, as it has a strong influence on the final number of clusters. Also, the performance of the proposed algorithm degrades when the dataset contains an excessively high number of images, presents complex features (i.e., numerous windows or glass walls) or is prone to visual aliasing.

The results might be considered successful since our incremental algorithm starts with a reduced number of images and updates the map (clusters) every time a new image arrives whereas the batch clustering algorithm has complete information on all the descriptors from the beginning of the process, and can calculate all the mutual similitudes to perform the clustering process. Additionally, gist proved to perform less robustly and the silhouettes obtained with batch spectral clustering are relatively higher. Therefore, the proposed incremental method along with omnidirectional images and the HOG descriptor constitutes the most suitable option to perform incremental mapping.

This work opens the door to new research works on incremental hierarchical map creation using global-appearance description methods in mobile robotics. Once we have tested the robustness and efficiency of the methods in real environments including human activity and presence of changes in the position of some objects, the next step is to improve the algorithm and adapt it to be used in large and outdoor environments. In this sense, we will focus in particular on the problem of abrupt changes of the lighting conditions and changes across seasons, as it is one of the issues that may have a more negative impact upon the visual mapping algorithms.

Supplementary Materials: The following are available online at http://www.mdpi.com/2076-3417/10/18/6480/s1, Table S1: General parameters of the global appearance descriptors, Table S2: Parameters that impact the size of the image descriptors; Table S3: Symbols used in the hierarchical incremental mapping processes, Table S4: Parameters that need to be tuned, Figure S1. Time of the process [s] using HOG descriptor depending on the number of images that it contains and γ parameter, Figure S2. Time of the process [s] using gist descriptor depending on the number of images that it contains and γ parameter.

Author Contributions: L.P. and Ó.R. conceived and designed the experiments; V.R. and S.C. performed the experiments; V.R., L.P. and S.C. analyzed the data; V.R. and L.P. implemented the necessary software. The paper was written and revised collaboratively by all the authors. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Generalitat Valenciana and the FSE through the grants ACIF/2018/224 and ACIF/2017/146, by the Spanish government through the project DPI 2016-78361-R (AEI/FEDER, UE): "Creación de mapas mediante métodos de apariencia visual para la navegación de robots", and by Generalitat Valenciana through the project AICO/2019/031: "Creación de modelos jerárquicos y localización robusta de robots móviles en entornos sociales".

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Colleens, T.; Colleens, J. Occupancy grid mapping: An empirical evaluation. In Proceedings of the 2007 Mediterranean Conference on Control & Automation, Athens, Greece, 27–29 June 2007; pp. 1–6.
2. Werner, S.; Krieg-Brückner, B.; Herrmann, T. Modelling navigational knowledge by route graphs. In *Spatial Cognition II*; Springer: Berlin, Germany, 2000; pp. 295–316.
3. Cebollada, S.; Payá, L.; Román, V.; Reinoso, O. Hierarchical localization in topological models under varying illumination using holistic visual descriptors. *IEEE Access* **2019**, *7*, 49580–49595. [CrossRef]
4. Kostavelis, I.; Charalampous, K.; Gasteratos, A.; Tsotsos, J.K. Robot navigation via spatial and temporal coherent semantic maps. *Eng. Appl. Artif. Intell.* **2016**, *48*, 173–187. [CrossRef]
5. Von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]
6. Grudic, G.Z.; Mulligan, J. Topological Mapping with Multiple Visual Manifolds. In *Robotics: Science and Systems*; MIT Press: Cambridge, MA, USA, 2005; pp. 185–192.
7. Valgren, C.; Lilienthal, A.J. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robot. Auton. Syst.* **2010**, *58*, 149–156.
8. Štimec, A.; Jogan, M.; Leonardis, A. Unsupervised learning of a hierarchy of topological maps using omnidirectional images. *Int. J. Pattern Recognit. Artif. Intell.* **2008**, *22*, 639–665. [CrossRef]
9. Payá, L.; Mayol, W.; Cebollada, S.; Reinoso, O. Compression of topological models and localization using the global appearance of visual information. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5630–5637.
10. Zivkovic, Z.; Bakker, B.; Krose, B. Hierarchical map building and planning based on graph partitioning. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA 2006), Orlando, FL, USA, 15–19 May 2006; pp. 803–809.
11. Cebollada, S.; Payá, L.; Mayol, W.; Reinoso, O. Evaluation of clustering methods in compression of topological models and visual place recognition using global appearance descriptors. *Appl. Sci.* **2019**, *9*, 377. [CrossRef]
12. Valgren, C.; Duckett, T.; Lilienthal, A. Incremental spectral clustering and its application to topological mapping. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 4283–4288.
13. Cha, Y.; Kim, D. *Omni-Directional Image Matching for Homing Navigation Based on Optical Flow Algorithm*; IEEE: Jeju Island, Korea, 2012;
14. Hata, A.; Wolf, D. *Outdoor Mapping Using Mobile Robots and Laser Range Finders*; IEEE: Cuernavaca, Mexico, 2009; pp. 209–214. [CrossRef]
15. Neto, L.B.; Grijalva, F.; Maike, V.R.M.L.; Martini, L.C.; Florencio, D.; Baranauskas, M.C.C.; Rocha, A.; Goldenstein, S. A Kinect-based wearable face recognition system to aid visually impaired users. *IEEE Trans. Hum.-Mach. Syst.* **2016**, *47*, 52–64. [CrossRef]
16. Häne, C.; Heng, L.; Lee, G.H.; Fraundorfer, F.; Furgale, P.; Sattler, T.; Pollefeys, M. 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image Vis. Comput.* **2017**, *68*, 14–27. [CrossRef]
17. Choi, J.; Ahn, S.; Choi, M.; Chung, W.K. Metric SLAM in home environment with visual objects and sonar features. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4048–4053.
18. Bonin-Font, F.; Ortiz, A.; Oliver, G. Visual navigation for mobile robots: A survey. *J. Intell. Robot. Syst.* **2008**, *53*, 263. [CrossRef]
19. Gálvez-López, D.; Salas, M.; Tardós, J.D.; Montiel, J. Real-time monocular object slam. *Robot. Auton. Syst.* **2016**, *75*, 435–449. [CrossRef]
20. Kriegman, D.J.; Triendl, E.; Binford, T.O. Stereo vision and navigation in buildings for mobile robots. *IEEE Trans. Robot. Autom.* **1989**, *5*, 792–803. [CrossRef]
21. Sturm, P.; Ramalingam, S.; Tardif, J.P.; Gasparini, S.; Barreto, J. Camera models and fundamental concepts used in geometric computer vision. *Found. Trends® Comput. Graph. Vis.* **2011**, *6*, 1–183.
22. Amorós, F.; Payá, L.; Marín, J.M.; Reinoso, O. Trajectory estimation and optimization through loop closure detection, using omnidirectional imaging and global-appearance descriptors. *Expert Syst. Appl.* **2018**, *102*, 273–290. [CrossRef]

23. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
24. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
25. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2010; pp. 778–792.
26. Angeli, A.; Doncieux, S.; Meyer, J.A.; Filliat, D. Visual topological SLAM and global localization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'09), Kobe, Japan, 12–17 May 2009; pp. 4300–4305.
27. Murillo, A.C.; Guerrero, J.J.; Sagues, C. Surf features for efficient robot localization with omnidirectional images. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3901–3907.
28. Gil, A.; Mozos, O.M.; Ballesta, M.; Reinoso, O. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Mach. Vis. Appl.* **2010**, *21*, 905–920. [CrossRef]
29. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]
30. Payá, L.; Reinoso, O.; Berenguer, Y.; Úbeda, D. Using omnidirectional vision to create a model of the environment: A comparative evaluation of global-appearance descriptors. *J. Sens.* **2016**, *2016*, 1209507. [CrossRef]
31. Payá, L.; Peidró, A.; Amorós, F.; Valiente, D.; Reinoso, O. Modeling environments hierarchically with omnidirectional imaging and global-appearance descriptors. *Remote Sens.* **2018**, *10*, 522. [CrossRef]
32. Oliva, A.; Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
33. Siagian, C.; Itti, L. Biologically inspired mobile robot vision localization. *IEEE Trans. Robot.* **2009**, *25*, 861–873. [CrossRef]
34. Zhou, X.; Su, Z.; Huang, D.; Zhang, H.; Cheng, T.; Wu, J. Robust Global Localization by Using Global Visual Features and Range Finders Data. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 218–223.
35. Menegatti, E.; Maeda, T.; Ishiguro, H. Image-based memory for robot navigation using properties of omnidirectional images. *Robot. Auton. Syst.* **2004**, *47*, 251–267. [CrossRef]
36. Radon, J. 1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Class. Pap. Mod. Diagn. Radiol.* **2005**, *5*, 21.
37. Berenguer, Y.; Payá, L.; Valiente, D.; Peidró, A.; Reinoso, O. Relative Altitude Estimation Using Omnidirectional Imaging and Holistic Descriptors. *Remote Sens.* **2019**, *11*, 323. [CrossRef]
38. Román, V.; Payá, L.; Reinoso, Ó. Evaluating the robustness of global appearance descriptors in a visual localization task, under changing lighting conditions. In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics, Porto, Portugal, 29–31 July 2018; pp. 258–265.
39. Xu, S.; Chou, W.; Dong, H. A Robust Indoor Localization System Integrating Visual Localization Aided by CNN-Based Image Retrieval with Monte Carlo Localization. *Sensors* **2019**, *19*, 249. [CrossRef] [PubMed]
40. Leyva-Vallina, M.; Strisciuglio, N.; Lopez-Antequera, M.; Tylecek, R.; Blaich, M.; Petkov, N. TB-Places: A Data Set for Visual Place Recognition in Garden Environments. *IEEE Access* **2019**, *7*, 52277–52287. [CrossRef]
41. Pantazi, X.E.; Tamouridou, A.A.; Alexandridis, T.; Lagopodi, A.L.; Kashefi, J.; Moshou, D. Evaluation of hierarchical self-organising maps for weed mapping using UAS multispectral imagery. *Comput. Electron. Agric.* **2017**, *139*, 224–230. [CrossRef]
42. Hagiwara, Y.; Inoue, M.; Kobayashi, H.; Taniguchi, T. Hierarchical spatial concept formation based on multimodal information for human support robots. *Front. Neurorobot.* **2018**, *12*, 11. [CrossRef]
43. Hwang, Y.; Choi, B.S. Hierarchical System Mapping for Large-Scale Fault-Tolerant Quantum Computing. *arXiv* **2018**, arXiv:1809.07998.
44. Balaska, V.; Bampis, L.; Boudourides, M.; Gasteratos, A. Unsupervised semantic clustering and localization for mobile robotics tasks. *Robot. Auton. Syst.* **2020**, *131*, 103567. [CrossRef]
45. Korrapati, H.; Mezouar, Y. Multi-resolution map building and loop closure with omnidirectional images. *Auton. Robot.* **2017**, *41*, 967–987. [CrossRef]

46. Latif, Y.; Huang, G.; Leonard, J.; Neira, J. Sparse optimization for robust and efficient loop closing. *Robot. Auton. Syst.* **2017**, *93*, 13–26. [CrossRef]
47. Carrasco, P.L.N.; Bonin-Font, F.; Oliver-Codina, G. Global image signature for visual loop-closure detection. *Auton. Robot.* **2016**, *40*, 1403–1417.
48. Payá, L.; Amorós, F.; Fernández, L.; Reinoso, O. Performance of global-appearance descriptors in map building and localization using omnidirectional vision. *Sensors* **2014**, *14*, 3033–3064. [CrossRef] [PubMed]
49. Román, V.; Payá, L.; Flores, M.; Cebollada, S.; Reinoso, Ó. Performance of New Global Appearance Description Methods in Localization of Mobile Robots. In *Iberian Robotics Conference*; Springer: Porto, Portugal, 2019; pp. 351–363.
50. Berenguer, Y.; Payá, L.; Ballesta, M.; Reinoso, O. Position estimation and local mapping using omnidirectional images and global appearance descriptors. *Sensors* **2015**, *15*, 26368–26395. [CrossRef] [PubMed]
51. Valiente, D.; Gil, A.; Reinoso, Ó.; Juliá, M.; Holloway, M. Improved omnidirectional odometry for a view-based mapping approach. *Sensors* **2017**, *17*, 325. [CrossRef]
52. Saito, M.; Kitaguchi, K. Appearance based robot localization using regression models. *IFAC Proc. Vol.* **2006**, *39*, 584–589. [CrossRef]
53. Pronobis, A.; Caputo, B. COLD: COsy Localization Database. *Int. J. Robot. Res. (IJRR)* **2009**, *28*, 588–594. [CrossRef]
54. Zhu, Q.; Yeh, M.C.; Cheng, K.T.; Avidan, S. Fast human detection using a cascade of histograms of oriented gradients. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1491–1498.
55. Hofmeister, M.; Liebsch, M.; Zell, A. Visual self-localization for small mobile robots with weighted gradient orientation histograms. In Proceedings of the 40th International Symposium on Robotics (ISR), Barcelona, Spain, 10–13 March 2009; pp. 87–91.
56. Hofmeister, M.; Vorst, P.; Zell, A. A comparison of efficient global image features for localizing small mobile robots. In Proceedings of the ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), Munich, Germany, 7–9 June 2010; pp. 1–8.
57. Torralba, A. Contextual priming for object detection. *Int. J. Comput. Vis.* **2003**, *53*, 169–191. [CrossRef]
58. Oliva, A.; Torralba, A. Building the gist of a scene: The role of global image features in recognition. *Prog. Brain Res.* **2006**, *155*, 23–36.
59. Chang, C.K.; Siagian, C.; Itti, L. Mobile robot vision navigation & localization using gist and saliency. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4147–4154.
60. Murillo, A.C.; Singh, G.; Kosecká, J.; Guerrero, J.J. Localization in urban environments using a panoramic gist descriptor. *IEEE Trans. Robot.* **2012**, *29*, 146–160. [CrossRef]
61. Liu, Y.; Zhang, H. Visual loop closure detection with a compact image descriptor. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 1051–1056.
62. Mahalanobis, P.C. *On the Generalized Distance in Statistics*; National Institute of Science of India: Jatani, India, 1936.

Bibliografía

[1] M. Agrawal, K. Konolige, and M. R. Blas. Censure: Center surround extremas for realtime feature detection and matching. In *European Conference on Computer Vision*, pages 102–115. Springer, 2008. 28

[2] M. Aladem and S. A. Rawashdeh. Lightweight visual odometry for autonomous mobile robots. *Sensors*, 18(9):2837, 2018. 29

[3] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *CVPR 2012: Computer Vision and Pattern Recognition*, pages 510–517, 2012. 22

[4] M. Alatise and G. Hancke. A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access*, 8:39830–39846, 2020. 13

[5] F. Amorós, L. Payá, J. M. Marín, and O. Reinoso. Trajectory estimation and optimization through loop closure detection, using omnidirectional imaging and global-appearance descriptors. *Expert Systems with Applications*, 102:273–290, 2018. i, 89, 90

[6] F. Amorós, L. Payá, W. Mayol-Cuevas, L. M. Jiménez, and O. Reinoso. Holistic descriptors of omnidirectional color images and their performance in estimation of position and orientation. *IEEE Access*, 8:81822–81848, 2020. 16, 22

[7] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3348–3353. IEEE, 2005. 28

[8] A. Angeli, S. Doncieux, J. Meyer, and D. Filliat. Visual topological slam and global localization. In *2009 IEEE International Conference on Robotics and Automation*, pages 4300–4305. IEEE, 2009. 15

[9] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 20, 118

[10] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Bronte. Fast and effective visual place recognition using binary codes and disparity information. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3089–3094. IEEE, 2014. 24, 29

[11] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit. Instant outdoor localization and slam initialization from 2.5 d maps. *IEEE transactions on visualization and computer graphics*, 21(11):1309–1318, 2015. 2

[12] ARVC. Automation, Robotics and Computer Vision Research Group. Miguel Hernández University. Spain. Quorum 5 set of images. accessed on 29th december 2020. http://arvc.umh.es/db/images/quorumv/. 38

[13] M. F. Aslan, A. Durdu, K. Sabanci, and M. A. Mutluer. Cnn and hog based comparison study for complete occlusion handling in human tracking. *Measurement*, page 107704, 2020. 25

[14] H. Badino, D. Huber, and T. Kanade. Real-time topometric localization. In *2012 IEEE International Conference on Robotics and Automation*, pages 1635–1642. IEEE, 2012. 28

[15] V. Balaska, L. Bampis, M. Boudourides, and A. Gasteratos. Unsupervised semantic clustering and localization for mobile robotics tasks. *Robotics and Autonomous Systems*, page 103567, 2020. 17

[16] T. Baldawi. Fireman robot: a prototype design. *International Journal of Applied Engineering Research*, 12(23):13255–13264, 2017. 13

[17] M. Ballesta, L. Payá, S. Cebollada, O. Reinoso, and F. Murcia. A cnn regression approach to mobile robot localization using omnidirectional images. *Applied Sciences*, 11(16):7521, 2021. 16, 120

[18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. 15, 22, 28

[19] Y. Berenguer, L. Payá, M. Ballesta, and O. Reinoso. Position estimation and local mapping using omnidirectional images and global appearance descriptors. *Sensors*, 15(10):26368–26395, 2015. 2, 16, 30, 117

[20] Y. Berenguer, L. Payá, A. Peidró, A. Gil, and O. Reinoso. Nearest position estimation using omnidirectional images and global appearance descriptors. In *Robot 2015: Second Iberian Robotics Conference*, pages 517–529. Springer, 2016. 24, 30

[21] Y. Berenguer, L. Payá, D. Valiente, A. Peidró, and O. Reinoso. Relative altitude estimation using omnidirectional imaging and holistic descriptors. *Remote Sensing*, 11(3):323, 2019. 15, 22, 30

[22] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart. State estimation for legged robots-consistent fusion of leg kinematics and imu. *Robotics*, 17:17–24, 2013. 1

[23] J. Bodner, H. Wykypiel, G. Wetscher, and T. Schmid. First experiences with the da vinci™ operating robot in thoracic surgery. *European Journal of Cardiothoracic surgery*, 25(5):844–851, 2004. 13

[24] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008. 14

[25] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018. 27, 118

[26] J. J. Cabrera, S. Cebollada, L. Payá, M. Flores, and O. Reinoso. A robust cnn training approach to address hierarchical localization with omnidirectional images. In *ICINCO*, pages 302–310, 2021. 125, 129, 141

[27] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010. 15, 22, 29

[28] L. Cao, J. Ling, and X. Xiao. Study on the influence of image noise on monocular feature-based visual slam based on ffdnet. *Sensors*, 20(17):4922, 2020. 17, 22, 117

[29] P. N. Carrasco, F. Bonin-Font, and G. Oliver-Codina. Global image signature for visual loop-closure detection. *Autonomous Robots*, 40(8):1403–1417, 2016. 18

[30] D. Cattaneo, M. Vaghi, A. L. Ballardini, S. Fontana, D. Sorrenti, and W. Burgard. Cmrnet: Camera to lidar-map registration. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1283–1289. IEEE, 2019. 14, 118

[31] S. Cebollada, L. Payá, M. Flores, A. Peidró, and O. Reinoso. A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. *Expert Systems with Applications*, page 114195, 2020. 118

[32] S. Cebollada, L. Payá, M. Flores, V. Román, A. Peidró, and O. Reinoso. A deep learning tool to solve localization in mobile autonomous robotics. In *ICINCO*, pages 232–241, 2020. 129

[33] S. Cebollada, L. Payá, W. Mayol, and O. Reinoso. Evaluation of clustering methods in compression of topological models and visual place recognition using global appearance descriptors. *Applied Sciences*, 9(3):377, 2019. 2, 16, 76, 79, 95

[34] S. Cebollada, L. Payá, D. Valiente, X. Jiang, and O. Reinoso. An evaluation between global appearance descriptors based on analytic methods and deep learning techniques for localization in autonomous mobile robots. *ICINCO 2*, pages 284–291, 2019. 142

[35] S. Cebollada, L. Payá, M. Flores, A. Peidró, and O. Reinoso. A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. *Expert Systems with Applications*, page 114195, 2020. 2, 14, 20, 23

[36] S. Cebollada, L. Payá, V. Román, and O. Reinoso. Hierarchical localization in topological models under varying illumination using holistic visual descriptors. *IEEE Access*, 7:49580–49595, 2019. 4, 75, 120

[37] Y. Cha and D. Kim. Omni-directional image matching for homing navigation based on optical flow algorithm. In *2012 12th International Conference on Control, Automation and Systems*, pages 1446–1451. IEEE, 2012. 1

[38] C. K. Chang, C. Siagian, and L. Itti. Mobile robot vision navigation and localization using gist and saliency. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4147–4154, 2010. 27

[39] J. Choi, S. Ahn, M. Choi, and W. K. Chung. Metric slam in home environment with visual objects and sonar features. In *2006 IEEE/RSJ international conference on intelligent robots and systems*, pages 4048–4053. IEEE, 2006. 14

[40] F. Chollet. *Deep learning with Python*, volume 361. Manning New York, 2018. 117

[41] F. Chollet. *Deep learning with Python*. Simon and Schuster, 2021. 20

[42] T. Colleens and J. Colleens. Occupancy grid mapping: An empirical evaluation. In *2007 Mediterranean Conference on Control & Automation*, pages 1–6. IEEE, 2007. 75

[43] B. Congram and T. Barfoot. Experimental comparison of visual and single-receiver gps odometry. *arXiv preprint arXiv:2106.02122*, 2021. 14

[44] B. Coors, A. Paul Condurache, and A. Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018. 22

[45] S. P. da Silva, R. V. da Nóbrega, A. Medeiros, L. Marinho, J. Almeida, and P. Reboucas Filho. Localization of mobile robots with topological maps and classification with reject option using convolutional neural networks in omnidirectional images. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018. 117

[46] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005. 16, 25

[47] X. Dong, X. Dong, J. Dong, and H. Zhou. Monocular visual-imu odometry: A comparative evaluation of detector–descriptor-based methods. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2471–2484, 2019. 23

[48] M. Dymczyk, I. Gilitschenski, J. Nieto, S. Lynen, B. Zeisl, and R. Siegwart. Landmarkboost: Efficient visualcontext classifiers for robust localization. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 677–684, Oct 2018. 4

[49] Q. Feng, H. Shum, and S. Morishima. Resolving hand-object occlusion for mixed reality with joint deep learning and model optimization. *Computer Animation and Virtual Worlds*, 31(4-5):e1956, 2020. 19, 117

[50] F. Foroughi, Z. Chen, and J. Wang. A cnn-based system for mobile robot navigation in indoor environments via visual localization with a small dataset. *World Electric Vehicle Journal*, 12(3):134, 2021. 17

[51] J. Fuentes, J. Ruiz, and J. Rendón. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015. 2

[52] E. Garcia-Fidalgo and A. Ortiz. Vision-based topological mapping and localization by means of local invariant features and map refinement. *Robotica*, 33(7):1446–1470, 2015. 117

[53] A. Gil, O. Martinez Mozos, M. Ballesta, and O. Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual slam. *Machine Vision and Applications*, 21(6):905–920, 2010. 15, 23

[54] R. Gonzalez, D. Apostolopoulos, and K. Iagnemma. Slippage and immobilization detection for planetary exploration rovers via machine learning and proprioceptive sensing. *Journal of Field Robotics*, 35(2):231–247, 2018. 4

[55] G. Z. Grudic and J. Mulligan. Topological mapping with multiple visual manifolds. In *Robotics: Science and Systems*, pages 185–192, 2005. 76

[56] G. Gu, J. Zou, R. Zhao, X. Zhao, and X. Zhu. Soft wall-climbing robots. *Science Robotics*, 3(25), 2018. 13

[57] K. Gwinner, R. Jaumann, E. Hauber, H. Hoffmann, C. Heipke, J. Oberst, G. Neukum, V. Ansan, J. Bostelmann, A. Dumke, et al. The high resolution stereo camera (hrsc) of mars express and its approach to science analysis and mapping for mars and its satellites. *Planetary and Space Science*, 126:93–138, 2016. 2

[58] Y. Hagiwara, M. Inoue, H. Kobayashi, and T. Taniguchi. Hierarchical spatial concept formation based on multimodal information for human support robots. *Frontiers in neurorobotics*, 12:11, 2018. 4, 17

[59] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. Pollefeys. 3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017. 2

[60] M. Hasegawa and S. Tabbone. A shape descriptor combining logarithmic-scale histogram of radon transform and phase-only correlation function. In *2011 International Conference on Document Analysis and Recognition*, pages 182–186. IEEE, 2011. 30

[61] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 122, 123, 144

[62] T. Hoang and S. Tabbone. A geometric invariant shape descriptor based on the radon, fourier, and mellin transforms. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2085–2088. IEEE, 2010. 30

[63] M. Hofmeister, M. Liebsch, and A. Zell. Visual self-localization for small mobile robots with weighted gradient orientation histograms. In *Proceedings of the 40th International Symposium on Robotics, pp. 87-91*. IFR, 2009. 25

[64] M. Hofmeister, P. Vorst, and A. Zell. A comparison of efficient global image features for localizing small mobile robots. In *Proceedings of the 41st International Symposium on Robotics*, pages 143–150, 2010. 25

[65] A. Holliday and G. Dudek. Scale-robust localization using general object landmarks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1688–1694. IEEE, 2018. 19, 117

[66] M. Horst and R. Möller. Visual place recognition for autonomous mobile robots. *Robotics*, 6(2):9, 2017. 24

[67] S. Hu, H. Shum, X. Liang, F. Li, and N. Aslam. Facial reshaping operator for controllable face beautification. *Expert Systems with Applications*, 167:114067, 2021. 19, 117

[68] Y. Hwang and B. Choi. Hierarchical system mapping for large-scale fault-tolerant quantum computing. *arXiv preprint arXiv:1809.07998*, 2018. 4, 17

[69] H. Ishiguro and S. Tsuji. Image-based memory of environment. In *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 2, pages 634–639 vol.2, 1996. 24

[70] Y. Jia, M. Li, L. An, and X. Zhang. Autonomous navigation of a miniature mobile robot using real-time trinocular stereo machine. In *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*, volume 1, pages 417–421 vol.1, 2003. 2, 14

[71] W. Jiang and W. Wang. Face detection and recognition for home service robots with end-to-end deep neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2232–2236. IEEE, 2017. 117

[72] M. Juliá, A. Gil, and O. Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33(4):427–444, 2012. 31

[73] J. Junior, A. Tommaselli, and M. Moraes. Calibration of a catadioptric omnidirectional vision system with conic mirror. *ISPRS Journal of Photogrammetry and Remote Sensing*, 113:97–105, 2016. 21

[74] Z. Kato, G. Nagy, M. Humenberger, and G. Csurka. Detecting low-rank regions in omnidirectional images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3682–3692, 2021. 15

[75] M. Keating, J. Zhang, C. Feider, S. Retailleau, R. Reid, A. Antaris, B. Hart, G. Tan, T. Milner, K. Miller, et al. Integrating the masspec pen to the da vinci surgical system for in vivo tissue analysis during a robotic assisted porcine surgery. *Analytical Chemistry*, 92(17):11535–11542, 2020. 13

[76] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier. A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes. *IEEE Transactions on Robotics*, 36(2):561–569, 2019. 17, 23, 117

[77] B. Kim, H. Joe, and S. Yu. High-precision underwater 3d mapping using imaging sonar for navigation of autonomous underwater vehicle. *International Journal of Control, Automation and Systems*, pages 1–10, 2021. 14

[78] S. Kim, C. Roh, S. Kang, and M. Park. Outdoor navigation of a mobile robot using differential gps and curb detection. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3414–3419. IEEE, 2007. 1

[79] H. Korrapati and Y. Mezouar. Multi-resolution map building and loop closure with omnidirectional images. *Autonomous Robots*, 41(4):967–987, 2017. 14, 17, 117

[80] I. Kostavelis, K. Charalampous, A. Gasteratos, and J. Tsotsos. Robot navigation via spatial and temporal coherent semantic maps. *Engineering Applications of Artificial Intelligence*, 48:173–187, 2016. 17, 75

[81] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 20, 118, 120, 122, 123, 144

[82] B. Krose, R. Bunschoten, S. Hagen, B. Terwijn, and N. Vlassis. Visual homing in enviroments with anisotropic landmark distrubution. In *Autonomous Robots, 23(3), 2007, pp. 231-245*, 2007. 22

[83] Y. Latif, G. Huang, J. Leonard, and J. Neira. Sparse optimization for robust and efficient loop closing. *Robotics and Autonomous Systems*, 93:13–26, 2017. 18

[84] K. Lee, J. Rao, and Y. Youn. Endoscopic thyroidectomy with the da vinci robot system using the bilateral axillary breast approach (baba) technique: our initial experience. *Surgical Laparoscopy Endoscopy & Percutaneous Techniques*, 19(3):e71–e75, 2009. 13

[85] S. Leutenegger, M. Chli, and R. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011. 22

[86] M. Leyva-Vallina, N. Strisciuglio, M. Lopez-Antequera, R. Tylecek, M. Blaich, and N. Petkov. Tb-places: A data set for visual place recognition in garden environments. *IEEE Access*, 7:52277–52287, 2019. 14, 23, 117, 120

[87] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov. Place recognition in gardens by learning visual representations: data set and benchmark analysis. In *International Conference on Computer Analysis of Images and Patterns*, pages 324–335. Springer, 2019. 120

[88] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov. Generalized contrastive optimization of siamese networks for place recognition. *arXiv preprint arXiv:2103.06638*, 2021. 4, 20, 120

[89] R. Li, Q. Liu, J. Gui, D. Gu, and H. Hu. Indoor relocalization in challenging environments with dual-stream convolutional neural networks. *IEEE Transactions on Automation Science and Engineering*, 15(2):651–662, 2017. 16, 118

[90] C. Liang, Y. Tie, L. Qi, and C. Bi. Deep vidar: Cnn based 360 panoramic video system for outdoor robot visual navigation and slam. In *Tenth International Conference on Digital Image Processing (ICDIP 2018)*, volume 10806, page 1080663. International Society for Optics and Photonics, 2018. 15

[91] J. Lin, J. Peng, Z. Hu, X. Xie, R. Peng, et al. Orb-slam, imu and wheel odometry fusion for indoor mobile robot localization and navigation. *Academic Journal of Computing & Information Science*, 3(1), 2020. 17, 22, 117

[92] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann. High-speed laser localization for mobile robots. *Robotics and autonomous systems*, 51(4):275–296, 2005. 1

[93] S. Liu, S. Li, L. Pang, J. Hu, H. Chen, and X. Zhang. Autonomous exploration and map construction of a mobile robot based on the tghm algorithm. *Sensors*, 20(2):490, 2020. 31

[94] W. Liu, Y. Mo, and J. Jiao. An efficient edge-feature constraint visual slam. In *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, pages 1–7, 2019. 20, 118

[95] Y. Liu and H. Zhang. Visual loop closure detection with a compact image descriptor. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1051–1056. IEEE, 2012. 27, 117

[96] D. Longo and G. Muscato. A modular approach for the design of the alicia3 climbing robot for industrial inspection. *Industrial Robot: An International Journal*, 2004. 13

[97] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. 15, 22

[98] Y. Lu and G. Lu. Deep unsupervised learning for simultaneous visual odometry and depth estimation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2571–2575. IEEE, 2019. 20, 118

[99] S. Luthardt, V. Willert, and J. Adamy. Llama-slam: Learning high-quality visual landmarks for long-term mapping and localization. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2645–2652. IEEE, 2018. 17, 22, 117

[100] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007. 76

[101] J. Ma and J. Zhao. Robust topological navigation via convolutional neural network feature and sharpness measure. *IEEE Access*, 5:20707–20715, 2017. 23

[102] L. Ma, J. Chen, et al. Using rgb image as visual input for mapless robot navigation. *arXiv preprint arXiv:1903.09927*, 2019. 20, 118

[103] Y. Ma, Q. Li, J. Xing, G. Huo, and Y. Liu. An intelligent object detection and measurement system based on trinocular vision. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(3):711–724, 2019. 14

[104] P. C. Mahalanobis. On the generalized distance in statistics. In *National Institute of Science of India*, 1936. 81

[105] L. B. Marinho, J. Almeida, J. Souza, V. Albuquerque, and P. Rebouças Filho. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Systems with Applications*, 72:1–17, 2017. 23

[106] O. Martinez Mozos, K. Nakashima, H. Jung, Y. Iwashita, and R. Kurazume. Fukuoka datasets for place categorization. *The International Journal of Robotics Research*, 38(5):507–517, 2019. g, 3

[107] E. Menegatti, T. Maeda, and H. Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251 – 267, 2004. 16, 22, 24

[108] E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello. Omnidirectional vision scan matching for robot localization in dynamic environments. *IEEE Transactions on Robotics*, 22(3):523–535, June 2006. 2

[109] E. Menegatti, M. Zocaratto, E. Pagello, and H. Ishiguro. Image-based monte carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004. 17, 23, 24

[110] M. Milford. Visual route recognition with a handful of bits. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012. 22

[111] W. Ming-yi, H. Li-le, L. Yu, and S. Chao. Mobile robot localization algorithm based on multi-sensor information fusion. *Journal of Measurement Science & Instrumentation*, 11(2), 2020. 17

[112] O. Moolan-Feroze and A. Calway. Predicting out-of-view feature points for model-based camera pose estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 82–88. IEEE, 2018. 20

[113] O. Moolan-Feroze, K. Karachalios, D. Nikolaidis, and A. Calway. Improving drone localisation around wind turbines using monocular model-based tracking. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7713–7719. IEEE, 2019. 20, 118

[114] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. In *Sensor devices and systems for robotics*, pages 253–276. Springer, 1989. 75

[115] A. C. Murillo, G. Singh, J. Kosecka, and J. J. Guerrero. Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics*, 29(1):146–160, 2013. 27

[116] A.C. Murillo, J.J. Guerrero, and C. Sagues. Surf features for efficient robot localization with omnidirectional images. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3901 –3907, april 2007. 2, 15, 17, 23, 28

[117] T. Naseer and W. Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1525–1530. IEEE, 2017. 14

[118] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli. Indoor localization of mobile robots through qr code detection and dead reckoning data fusion. *IEEE/ASME Transactions On Mechatronics*, 22(6):2588–2599, 2017. 17

[119] D.I Neumann, T. Langner, F. Ulbrich, D. Spitta, and D. Goehring. Online vehicle detection using haar-like, lbp and hog feature based image classifiers with stereo vision preselection. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 773–778. IEEE, 2017. 25

[120] N. Nozawa, H. Shum, Q. Feng, E. Ho, and S. Morishima. 3d car shape reconstruction from a contour sketch using gan and lazy learning. *The Visual Computer*, pages 1–14, 2021. 19, 117

[121] K. Okuyama, T. Kawasaki, and V. Kroumov. Localization and position correction for mobile robot using artificial visual landmarks. In *Advanced Mechatronic Systems (ICAMechS), 2011 International Conference on*, pages 414–418, 2011. 2

[122] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. In *International Journal of Computer Vision, Vol. 42(3): 145-175.*, 2001. 16, 22, 24

[123] A. Oliva and A. Torralba. Building the gist of ascene: the role of global image features in recognition. In *Progress in Brain Reasearch: Special Issue on Visual Perception.Vol. 155*, 2006. 27

[124] D. Organisciak, D. Sakkos, E. Ho, N. Aslam, and H. Shum. Unifying person and vehicle re-identification. *IEEE Access*, 8:115673–115684, 2020. 19, 117

[125] X. E. Pantazi, A. A. Tamouridou, TK Alexandridis, A. L. Lagopodi, J. Kashefi, and D. Moshou. Evaluation of hierarchical self-organising maps for weed mapping using uas multispectral imagery. *Computers and Electronics in Agriculture*, 139:224–230, 2017. 4, 17

[126] L. Payá, F. Amorós, L. Fernández, and O. Reinoso. Performance of global-appearance descriptors in map building and localization using omnidirectional vision. *Sensors*, 14(2):3033–3064, 2014. 9, 17, 23, 78

[127] L. Payá, L. Fernández, A. Gil, and O. Reinoso. Map building and monte carlo localization using global appearance of omnidirectional images. *Sensors*, 10(12):11468–11497, 2010. 25, 28

[128] L. Payá, A. Gil, and O. Reinoso. A state-of-the-art review on mapping and localization of mobile robots using omnidirectional vision sensors. *Journal of Sensors*, 2017, 2017. 2

[129] L. Payá, W. Mayol, S. Cebollada, and O. Reinoso. Compression of topological models and localization using the global appearance of visual information. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5630–5637. IEEE, 2017. 76

[130] L. Payá, A. Peidró, F. Amorós, D. Valiente, and O. Reinoso. Modeling environments hierarchically with omnidirectional imaging and global-appearance descriptors. *Remote Sensing*, 10(4):522, 2018. 16

[131] L. Paya, O. Reinoso, Y. Berenguer, and D Ubeda. Using omnidirectional vision to create a model of the environment: a comparative evaluation of global appearance descriptors. *Journal of Sensors*, 2016:1–21, 2016. 16, 23, 24, 31

[132] G. Pintore, M. Agus, and E. Gobbetti. AtlantaNet: Inferring the 3D indoor layout from a single 360 image beyond the Manhattan World assumption. In *Proc. ECCV*, pages 432–448, 2020. 22

[133] T. Pire, T. Fischer, J. Civera, P. De Cristóforis, and J. J. Berlles. Stereo parallel tracking and mapping for robot localization. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1373–1378. IEEE, 2015. 2

[134] R. Polvara, S. Sharma, J. Wan, A. Manning, and R. Sutton. Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles. *The Journal of Navigation*, 71(1):241–256, 2018. 19, 117

[135] A. Pronobis and B. Caputo. COLD: COsy Localization Database. *The International Journal of Robotics Research (IJRR)*, 28(5):588–594, May 2009. e, 10, 70, 89, 91, 118, 129, 130, 142

[136] X. Qin, G. Wu, J. Lei, F. Fan, X. Ye, and Q. Mei. A novel method of autonomous inspection for transmission line based on cable inspection robot lidar data. *Sensors*, 18(2):596, 2018. 13

[137] J. Radon. 1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Classic papers in modern diagnostic radiology*, 5:21, 2005. 16, 30

[138] O. Reinoso and L. Payá. Special issue on mobile robots navigation. *Applied Sciences*, 10(4):1317, 2020. 14, 21

[139] O. Reinoso and L. Payá. Special issue on visual sensors. *Sensors*, 20(3):910, 2020. 14, 21, 117

[140] V. Roman, S. Cebollada, L. Payá, M. Flores, A. Gil, and O. Reinoso. Evaluación de nuevos modos de empleo de los descriptors de apariencia global en tareas de localización. In *XL JORNADAS DE AUTOMATICA (El Ferrol (Spain), 4-6 September 2019)*, pages 842–848. Ed. CEA-IFAC, 2019. 8

[141] V. Román, L. Payá, S. Cebollada, A. Peidró, and O. Reinoso. Evaluating the robustness of new holistic description methods in position estimation of mobile robots. In *International Conference on Informatics in Control, Automation and Robotics*, pages 207–225. Springer, 2022. 7, 74

[142] V. Román, L. Payá, S. Cebollada, and O. Reinoso. Creating incremental models of indoor environments through omnidirectional imaging. *Applied Sciences*, 10(18):6480, 2020. 7, 17, 23, 115, 153

[143] V. Román, L. Payá, A. Peidró, D. Valiente, L. M. Jiménez, and O. Reinoso. Evaluación de descriptores de apariencia global en tareas de localización bajo cambios de iluminación. In *XXXIX Jornadas de Automática*, pages 306–313. Área de Ingeniería de Sistemas y Automática, Universidad de Extremadura, 2018. 8

[144] V. Román, L. Payá, S. Cebollada, A. Peidró, and O Reinoso. An evaluation of new global appearance descriptor techniques for visual localization in mobile robots under changing lighting conditions. In *ICINCO 2020, 17th International Conference on Informatics in Control, Automation and Robotics (Lieusaint-Paris, France, 7-9 July, 2020)*. Ed. INSTICC, 2020. 7, 74

[145] V. Román, L. Payá, M. Flores, S. Cebollada, and O. Reinoso. Performance of new global appearance description methods in localization of mobile robots. In *Iberian Robotics conference*, pages 351–363. Springer, 2019. 7, 74, 78

[146] V. Román, L. Payá, A. Peidró, M. Ballesta, and O. Reinoso. The role of global appearance of omnidirectional images in relative distance and orientation retrieval. *Sensors*, 21(10):3327, 2021. 7, 74, 142, 153

[147] V. Román, L. Payá, and O. Reinoso. Evaluating the robustness of global appearance descriptors in a visual localization task, under changing lighting conditions. In *ICINCO-RA*, pages 258–265, 2018. 8, 74, 79

[148] X. Ruan, D. Ren, X. Zhu, and J. Huang. Mobile robot navigation based on deep reinforcement learning. In *2019 Chinese control and decision conference (CCDC)*, pages 6174–6178. IEEE, 2019. 14, 117

[149] F. Rubio, F. Valero, and C. Llopis-Albert. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2), 2019. 13

[150] E. Rublee, V. Rabud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV 2011: International Conference on Computer Vision*, pages 2564–2571, 2011. 22

[151] D. Sakkos, E. Ho, H. Shum, and G. Elvin. Image editing-based data augmentation for illumination-insensitive background subtraction. *Journal of Enterprise Information Management*, 2020. 118

[152] D. Sakkos, H. Shum, and E. Ho. Illumination-based data augmentation for robust background subtraction. In *2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pages 1–8. IEEE, 2019. 125, 141

[153] F. Shamsfakhr, B. Bigham, and A. Mohammadi. Indoor mobile robot localization in dynamic and cluttered environments using artificial landmarks. *Engineering Computations*, 2019. 17, 22, 117

[154] P. Sharma, H. Liu, H. Wang, and S. Zhang. Securing wireless communications of connected vehicles with artificial intelligence. In *2017 IEEE international symposium on technologies for homeland security (HST)*, pages 1–7. IEEE, 2017. 19, 117

[155] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, Feb 2007. 27, 28

[156] C. Siagian and L. Itti. Biologically inspired mobile robot vision localization. *IEEE Transactions on Robotics*, 25(4):861–873, 2009. 16, 17, 23

[157] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011. 13

[158] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 20, 118, 122, 123, 144

[159] H. Sinha, J. Patrikar, E. G. Dhekane, G. Pandey, and M. Kothari. Convolutional neural network based sensors for mobile robot relocalization. In *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, pages 774–779. IEEE, 2018. 20, 118

[160] K. Sommer, K. Kim, Y. Kim, and S. Jo. Towards accurate kidnap resolution through deep learning. In *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 2017. 20

[161] volume=22 number=04 pages=639–665 year=2008 publisher=World Scientific Stimec, A. and Jogan, M. and Leonardis, A., journal=International Journal of Pattern Recognition and Artificial Intelligence. Unsupervised learning of a hierarchy of topological maps using omnidirectional images. 4, 76

[162] W. Sturzl and H.A. Mallot. Efficient visual homing based on fourier transformed panoramic images. *Robotics and Autonomous Systems*, 54(4):300–313, 2006. 24

[163] Z. Su, X. Zhou, T. Cheng, H. Zhang, B. Xu, and W. Chen. Global localization of a mobile robot using lidar and visual features. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2377–2383. IEEE, 2017. 27

[164] C. Sun, C. Hsiao, M. Sun, and H. Chen. HorizonNet: Learning room layout with 1D representation and pano stretch data augmentation. In *Proc. CVPR*, June 2019. 22

[165] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020. 137

[166] N. Sünderhauf and P. Protzel. Brief-gist-closing the loop by simple means. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1234–1241. IEEE, 2011. 29

[167] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 20, 118

[168] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4297–4304. IEEE, 2015. 20

[169] J. Tang, Y. Ren, and S. Liu. Real-time robot localization, vision, and speech recognition on nvidia jetson tx1. *arXiv preprint arXiv:1705.10945*, 2017. 14

[170] G. Tanzmeister, J. Thomas, D. Wollherr, and M. Buss. Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6090–6095. IEEE, 2014. 19, 117

[171] J. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2531–2538. IEEE, 2008. 2

[172] A. Torralba. Contextual priming for object detection. *International journal of computer vision*, 53(2):169–191, 2003. 27

[173] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *IEEE International Conference on Robotics and Automation*, pages 1023–1029, 2000. 22

[174] C. Valgren, T. Duckett, and A. Lilienthal. Incremental spectral clustering and its application to topological mapping. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4283–4288. IEEE, 2007. 18, 76

[175] C. Valgren and A. Lilienthal. Sift, surf & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149–156, 2010. 4, 15, 76

[176] D. Valiente, A Gil, O. Reinoso, M. Juliá, and M. Holloway. Improved omnidirectional odometry for a view-based mapping approach. *Sensors*, 17(2):325, 2017. 2

[177] J. Wang, B. Tao, Z. Gong, W. Yu, and Z. Yin. A mobile robotic 3-d measurement method based on point clouds alignment for large-scale complex surfaces. *IEEE Transactions on Instrumentation and Measurement*, 2021. 14

[178] Y. Wang, T. Bao, C. Ding, and M. Zhu. Face recognition in real-world surveillance videos with deep learning method. In *2017 2nd international conference on image, vision and computing (icivc)*, pages 239–243. IEEE, 2017. 19, 117

[179] P. Weinzaepfel, G. Csurka, Y. Cabon, and M. Humenberger. Visual localization by learning objects-of-interest dense match regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5634–5643, 2019. 19, 118

[180] S. Werner, B. Krieg-Bruckner, and T. Herrmann. Modelling navigational knowledge by route graphs. In *Spatial cognition II*, pages 295–316. Springer, 2000. 75

[181] S. Xu, W. Chou, and H. Dong. A robust indoor localization system integrating visual localization aided by cnn-based image retrieval with monte carlo localization. *Sensors*, 19(2):249, 2019. 16, 23, 117, 120

[182] X. Yang and K. T. Cheng. Local difference binary for ultrafast and distinctive feature description. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):188–194, 2013. 22

[183] K. M. Yi, Y. Verdie, P. Fua, and V. Lepetit. Learning to assign orientations to feature points. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 107–116, 2016. 20

[184] Z. Yong-guo, C. Wei, and L. Guang-liang. The navigation of mobile robot based on stereo vision. In *Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on*, pages 670–673. IEEE, 2012. 2

[185] X. Yuan, J. Martínez-Ortega, J. Fernández, and M. Eckert. Aekf-slam: a new algorithm for robotic underwater navigation. *Sensors*, 17(5):1174, 2017. 17, 22

[186] H. F. Zaki, F. Shafait, and A. Mian. Viewpoint invariant semantic object and scene categorization with rgb-d sensors. *Autonomous Robots*, 43(4):1005–1022, 2019. 117

[187] M. Zhang, S. Han, S. Wang, X. Liu, M. Hu, and J. Zhao. Stereo visual inertial mapping algorithm for autonomous mobile robot. In *2020 3rd International Conference on Intelligent Robotic and Control Engineering (IRCE)*, pages 97–104. IEEE, 2020. 29

[188] Q. Zhao, B. Zhang, S. Lyu, H. Zhang, D. Sun, G. Li, and W. Feng. A cnn-sift hybrid pedestrian navigation method based on first-person vision. *Remote Sensing*, 10(8):1229, 2018. 20, 118

[189] X. Zhou, Z. Su, D. Huang, H. Zhang, T. Cheng, and J. Wu. Robust global localization by using global visual features and range finders data. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 218–223. IEEE, 2018. 16

[190] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498, 2006. 25

[191] Z. Zivkovic, B. Bakker, and B. Krose. Hierarchical map building and planning based on graph partitioning. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 803–809. IEEE, 2006.