TESIS DOCTORAL

Generación de vistas panorámicas y localización de un robot móvil mediante un sistema con un campo de visión de 360º

María Flores Tenza

2023

DIRECTOR:
Luis Payá Castelló

CODIRECTOR:
David Valiente García

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

Programa de Doctorado en
TECNOLOGÍAS INDUSTRIALES Y DE TELECOMUNICACIÓN

La presente Tesis Doctoral, titulada “Generación de vistas panorámicas y localización de un robot móvil mediante un sistema con un campo de visión de 360°”, se presenta bajo la modalidad de **tesis por compendio** de las siguientes **publicaciones**:

- Efficient probability-oriented feature matching using wide field-of-view imaging.
  María Flores, David Valiente, Arturo Gil, Oscar Reinoso y Luis Payá
  Engineering Applications of Artificial Intelligence, vol. 107, p. 104539, 2022.
  Ed. Elsevier
  DOI: 10.1016/j.engappai.2021.104539.
  Factor de impacto JCR 2022: 8.0
  Ranking JCR 2022. Categoría “ENGINEERING, MULTIDISCIPLINARY”: posición 5/90.
  Primer cuartil (Q1).

OTRAS PUBLICACIONES DE ESTA TESIS

- Generating a full spherical view by modelling the relation between two fisheye images
  María Flores, David Valiente, Adrián Peidró, Oscar Reinoso y Luis Payá
  Visual Computer (actualmente en evaluación)
  Ed. Springer
  Factor de impacto JCR 2022: 3.5
  Ranking JCR 2022. Categoría “COMPUTER SCIENCE, SOFTWARE ENGINEERING”: posición 34/108. Segundo cuartil (Q2).

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

INFORMA:

Que Dña. María Flores Tenza ha realizado bajo la supervisión de nuestro Programa de Doctorado el trabajo titulado **"Generación de vistas panorámicas y localización de un robot móvil mediante un sistema con un campo de visión de 360º"** 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 a de de 2023

Prof. Dr. D. Óscar Reinoso García

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

El Dr. D. Luis Payá Castelló, director/a, y el Dr. D. David Valiente García, codirector/a de la tesis doctoral titulada **“Generación de vistas panorámicas y localización de un robot móvil mediante un sistema con un campo de visión de 360º”**

INFORMAN:

Que Dña. María Flores Tenza ha realizado bajo nuestra supervisión el trabajo titulado **“Generación de vistas panorámicas y localización de un robot móvil mediante un sistema con un campo de visión de 360º”** 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 a de de 2023

Director/a de la tesis	Codirector/a de la tesis
Dr. D. Luis Payá Castelló	Dr. D. David Valiente García

Abstract

Nowadays, mobile robots operate in several fields, not only in the industry or in those which are dangerous to humans (such as planetary or mine explorations), but also in our daily lives (such as in restaurants or even at our homes). In many situations, they must carry out a specific task while navigating autonomously. To achieve this safely, the mobile robot must know its environment sufficiently and be able to localize itself within it.

Following the above paragraph, the autonomous navigation involves solving a set of tasks such as localization, mapping or planning trajectories while avoiding obstacles. Information about the environment must be available for the mobile robot during the navigation. Therefore, the mobile robot must carry certain sensors on board based on the required amount and type of information. Among all the sensors used in the mobile robotics field, the vision systems are being widely employed in order to achieve autonomous navigation. The reason is that a unique image is capable of providing a wide variety of information (e.g. texture or color) and it can be employed to solve the localization problem. Furthermore, they are a suitable solution for different environment types (aquatic, aerial, terrestrial or indoor/outdoor). In terms of amount of information, the omnidirectional vision systems are capable of providing a field of view of 360° around the mobile robot.

The present thesis has two main objectives. On the one hand, some contributions to improve a probabilistic localization algorithm are proposed to achieve a more robust estimation of the relative pose from a pair of wide field-of-view images. On the other hand, an algorithm is presented to generate a full spherical view from a pair of fisheye images, including some correction steps that improve the quality of the resulting image. These two objectives are described below in more detail.

Regarding the first objective, the localization task can be addressed as a visual odometry problem. There are different approaches to carry out this, but the one chosen in this thesis is the approach based on local characteristic points. In this case, the algorithm has implemented a search of feature matches between the image captured at the present instant time and the one captured at the previous instant time. Then, the set of local feature points is employed to estimate the essential matrix, which can be decomposed into a rotation matrix and a translation vector (except for a scale factor). In this work, we propose to implement a matching search based on a probabilistic model of the environment to improve the visual odometry algorithm and we perform a comparative evaluation of wide-field-of-view vision systems in this task.

The second objective refers to the vision system used in this work, which is composed of two back-to-back CMOS sensors and two fisheye lenses with a field of view greater than 180° each. This configuration permits generating a whole field of view, 360° horizontally and 180° vertically. The latter is a valuable advantage, especially for autonomous navigation. However, it also requires precise processing to blend both images and create a complete panoramic view of the environment and it may lead to

errors and artifacts in the resulting image, mainly in the overlapping areas. In view of the above, this work aims to minimize the alignment error before calculating the geometric transformation through several contributions which include some correction steps. The experimental section proves that they are effective to generate a high-quality panoramic view.

Resumen

Actualmente, los robots móviles están presentes en varios ámbitos, no solo en la industria o en aquellos que presentan algún peligro para el ser humano (como ocurre en las exploraciones en planetas o en minas), sino también en nuestro día a día (como en restaurantes o incluso en nuestros hogares). En muchas ocasiones, deben realizar la tarea asignada mientras navegan de forma autónoma. Para lograrlo de forma segura, el robot móvil debe conocer bien su entorno y ser capaz de localizarse en él.

En línea con el párrafo anterior, la navegación autónoma implica resolver una serie de tareas como localización, creación de mapas o planificación de trayectorias evitando obstáculos. Durante la navegación, el robot debe tener información sobre su entorno. Por consiguiente, el robot móvil debe llevar a bordo ciertos sensores en función de la cantidad y tipo de información que necesite. De entre todos los sensores utilizados en el campo de la robótica móvil, los sistemas de visión están siendo ampliamente empleados para conseguir una navegación autónoma. La razón es que una imagen única es capaz de proporcionar una gran variedad de información (por ejemplo, textura o color) y puede emplearse para resolver el problema de localización. Además, son una solución adecuada para distintos tipos de entorno (como acuático, aéreo o terrestre, ya sea en interior o exterior). En cuanto a cantidad de información se refiere, los sistemas de visión omnidireccionales son capaces de proporcionar un campo de visión de 360° alrededor del robot móvil.

Tras todo lo comentado, la presente tesis tiene dos objetivos principales. Por un lado, se proponen algunas contribuciones para mejorar un algoritmo de localización probabilística para conseguir una estimación más robusta de la pose relativa a partir de un par de imágenes de campo de visión amplio. Por otro lado, se presenta un algoritmo para generar una vista esférica completa a partir de un par de imágenes *fisheye*, incluyendo algunos pasos de corrección que mejoran la calidad de la imagen resultante. Ambos objetivos se describen con más detalle a continuación.

Respecto al primer objetivo, la tarea de localización puede abordarse como un problema de odometría visual. Hay varios procedimientos para ello, pero nos centraremos en el basado en puntos característicos locales. En este caso, el algoritmo ha implementado una búsqueda de coincidencias de características entre la imagen capturada en el instante de tiempo actual y la capturada en el instante de tiempo anterior. A continuación, el conjunto de puntos característicos locales se emplea para estimar la matriz esencial, que puede descomponerse en una matriz de rotación y un vector de traslación (excepto un factor de escala). En este trabajo, proponemos implementar una búsqueda de coincidencias basada en un modelo probabilístico del entorno para mejorar el algoritmo de odometría visual y realizamos una evaluación comparativa de sistemas de visión de campo amplio en esta tarea.

El segundo objetivo se refiere al sistema de visión utilizado en este trabajo, que está compuesto por dos sensores CMOS *back-to-back* y dos lentes *fisheye* con un campo de visión superior a 180° cada una. Esta configuración permite generar un campo de visión

completo, de 360° horizontalmente y 180° verticalmente. Esto último constituye una valiosa ventaja, especialmente para la navegación autónoma. Sin embargo, también requiere un procesamiento preciso para mezclar ambas imágenes y crear una vista panorámica completa del entorno, y puede dar lugar a errores y artefactos en la imagen resultante, principalmente en las zonas de solape. Teniendo en cuenta lo anterior, este trabajo pretende minimizar el error de alineación antes de calcular la transformación geométrica mediante varias contribuciones que incluyen algunos pasos de corrección. La parte experimental demuestra que son eficaces para generar una panorámica de alta calidad.

Agredecimientos

Me gustaría comenzar dándoles las gracias a mis directores de tesis, Luis Payá y David Valiente, porque sin su apoyo y tiempo esto no hubiese sido posible. Gracias por haber compartido vuestro conocimiento conmigo durante estos años, pues no solo se ha visto enriquecida esta tesis sino también me ha ayudado a crecer un poquito como investigadora. Muchas gracias por todo.

Agradecer a todos los miembros del grupo ARVC: Óscar, Luis Miguel, José María, Arturo, David Úbeda, Mónica y Adrián, por todo el apoyo y consejos que he recibido de vuestra parte. También he tenido la suerte de impartir docencia al lado de algunos de vosotros (Óscar, Mónica, Adrián, Luis Payá y David Valiente), lo cual ha sido una experiencia muy gratificante porque he aprendido mucho de cada uno y me gustaría agradeceros la ayuda y la paciencia que habéis tenido conmigo también en este ámbito.

Me gustaría extender estos agradecimientos a otra parte importante durante este camino, mis compañeros de laboratorio. Ha sido un camino largo por lo que he tenido la suerte de poder conocer y trabajar junto a muchas personas de las que también he aprendido. En primer lugar, me gustaría empezar dándoles las gracias a Vicente y a Sergio porque, desde el primer día que entré en el laboratorio, siempre me habéis ayudado y aconsejado. Además, gracias por los buenos momentos. Me siento muy afortunada de haber compartido laboratorio con vosotros. A Julio, aunque hemos coincidido menos tiempo, gracias por haber formado parte y por los momentos que hemos compartido. A Orlando, ha sido un placer haber sido tu compañera tanto en el grado como en esta etapa. A Juanjo, porque siempre he tenido tu apoyo y ayuda (que no ha sido poca). Gracias también por todos los momentos que hemos vivido y ojalá sean muchos más. Me alegro mucho de que hayas formado parte de este camino.

También a Álvaro, Antonio, Enrique, Fran Martínez, Fran Soler, Marc, Marcos, Mario, Miriam y Paula, por vuestro apoyo durante este último tramo y por todos los momentos que he compartido con vosotros. Me alegro mucho de haberos conocido a todos.

Agradecer a Helder Araujo por todo el tiempo que me dedicó durante mi estancia en el Instituto de Sistemas y Robótica (ISR) de la Universidad de Coimbra. Gracias por todas las ideas y por la ayuda que me proporcionó pues ello también ha enriquecido el desarrollo de esta tesis.

Fuera de la universidad, quiero dar las gracias a mi familia. A mi madre y a mi padre por ofrecerme vuestro apoyo incondicional siempre. Parte de lo que soy y estoy consiguiendo es gracias a vosotros. Gracias por ser como sois conmigo, por preocuparos por mí y por haberme dado todo el amor del mundo. También agradecer a mis hermanos, Olga y José Antonio. Gracias por haberme apoyado siempre, por haberme dado tanto amor y sobre todo por haber incluido a esta pequeña en muchos de vuestros planes. Por último a mis tres sobrinos. A Fran y José Alberto por todos los momentos que hemos vivido desde pequeños. A Gonzalo, porque eres el pequeño de esta familia y con tu sonrisa y energía nos alegras a todos.

A Cristina, por todos los momentos que hemos pasado juntas, por estar ahí siempre y porque sé que te alegras mucho de mis logros. Que esta amistad que empezó hace mucho tiempo nunca se acabe, porque eres imprescindible en mi vida. A Clara, por tu apoyo y por todo lo que hemos vivido desde que nos conocimos en la Universidad, porque eres muy importante para mí.

Por último, pero no menos importante a Manuel. Gracias por ser mi apoyo siempre, por escucharme cuando lo necesito y darme tu punto de vista que siempre suele ser más positivo que el mío. Gracias por confiar en mí y animarme. Nunca olvides que estoy muy orgullosa de ti por todo lo que has logrado (y lograrás) y que siempre tendrás mi apoyo. Hace años decidimos ser compañeros de vida, por lo que a tu lado he crecido y he compartido muchas etapas de mi vida y me siento muy agradecida por ello. Gracias por todo lo que tenemos juntos, porque tú y Klaus sois mi felicidad diaria. También agradecer a tu familia por haberme acogido como una más. A tus padres, Enrique y Manuela, gracias por apoyarme, ayudarme y estar ahí siempre que lo necesito. Gracias por tratarme como una hija. A tu hermano, Enrique, gracias por todos los momentos que hemos vivido y por estar siempre ahí.

La presente tesis ha sido cofinanciada por la Generalitat Valenciana y el Fondo Social Europeo (FSE) a través de la subvención predoctoral, con referencia ACIF/2020/141, incluida en el Programa Operativo del Fondo Social Europeo 2014-2020 de la Comunitat Valenciana.

Financiación

La realización de la presente tesis doctoral ha tenido lugar en un marco que ha recibido diferente financiación tanto de becas como de colaboraciones y proyectos de investigación competitivos.

Becas

En particular, el desarrollo de las distintas investigaciones que se incluyen en la tesis ha sido posible gracias a la obtención de la beca ACIF (referencia: ACIF/2020/141), de la Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital, cofinanciada, a su vez, por fondos de la Unión Europea a través del Programa Operativo del Fondo Social Europeo (European Social Fund, ESF). De esta forma, la beca referida ha permitido que la autora de la tesis disponga de financiación durante un periodo de tres años (del 01/10/2020 al 30/09/2023) y, por tanto, lleve a cabo el estudio propuesto.

Por otro lado, cabe destacar la relevancia de otras becas y programas en relación con la movilidad internacional, pues han favorecido la realización de una estancia en una universidad de prestigio situada en el extranjero. En el año 2021, la autora de esta tesis realizó una estancia de investigación de tres meses de duración (del 20 de septiembre al 21 de diciembre) en el Instituto de Sistemas y Robótica de la Universidad de Coimbra en Portugal. Dicha estancia fue financiada por la Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital junto con el Fondo Social Europeo mediante la subvención con referencia BEFPI/2021/027.

Proyectos de investigación

En el periodo de realización de la presente tesis doctoral, la autora ha participado en diferentes proyectos de investigación, que a continuación se detallan:

- "Creación de Mapas Mediante Métodos de Apariencia Visual para la Navegación de Robots", financiado por CICYT Ministerio de Ciencia e Innovación, con referencia DPI2016-78361-R. Duración: 01/01/2017 al 31/12/2019
- "Creación de modelos jerárquicos y localización robusta de robots móviles en entornos sociales", financiado por la Generalitat Valenciana, con referencia AICO/2019/031. Duración: 01/01/2019 a 31/12/2020
- "Robots híbridos y reconstrucción multisensorial para aplicaciones en estructuras reticulares (HyReBot)", financiado por el Ministerio de Ciencia e Innovación (PID2020-116418RB-I00). Duración: 09/2021 - 08/2024
- "Hacia una mayor integración de robots inteligentes en la sociedad: navegar, reconocer y manipular", financiado por la Generalitat Valenciana, con referencia PROMETEO/2021/075. Duración: 01/2021 - 12/2024
- "Desarrollo de tecnologías móviles inteligentes para tareas de seguridad y vigilancia de entornos de interior y exterior, financiado por la Agencia Estatal de Investigación. Ministerio de Ciencia e Innovación, con referencia TED2021-130901B-I00. Duración: 12/2022 - 11/2024

NOMENCLATURA EMPLEADA EN LAS ECUACIONES

$\boldsymbol{X}$ Punto en el espacio 3D (i.e. $\boldsymbol{X} \in \mathbb{R}^3$): letra en mayúscula, cursiva y negrita.

$\boldsymbol{x}$ Punto en el espacio 2D (i.e. $\boldsymbol{x} \in \mathbb{R}^2$): letra en minúscula, cursiva y negrita.

$\mathbf{X}$ Matriz: letra en mayúscula, negrita y sin cursiva.

$\mathbf{x}$ Vector columna o fila: letra en minúscula, negrita y sin cursiva.

$\vec{X}$ Segmento dirigido en el espacio tridimensional: letra en mayúscula y cursiva.

$\vec{x}$ Segmento dirigido en el plano: letra en minúscula y cursiva.

x Escalar: letra en cursiva.

ACRÓNIMOS

AP *Average Precision*

APOFM *Adaptive Probability-Oriented Feature Matching* [1]

BCM *Bayesian Committee Machine*

BoW *Bag of Visual Words*

BRIEF *Binary Robust Independent Elementary Features*) [2]

CCD *Charge-Coupled Device*

CMOS *Complementary Metal Oxide Semiconductor*

CNN *Convolutional Neural Network*

DSCM Double Sphere Camera Model [3]

EUCM *Extended Unified Camera Model* [4]

FAST *Features from Accelerated Segment Test* [5]

FOV *Field Of View*

GNSS *Global Navigation Satellite System*

GP *Gaussian Process* [6]

GPS *Global Positioning System*

HOG *Histogram of Oriented Gradients*

ICP *Iterative Closest Point*

IMU *Inertial Measurement Units*

INS *Inertial Navigation System*

IQ *Image Quality*

k-nn *k-nearest neighbors*

KB Kannala-Brand [7]

KLT Kanade–Lucas–Tomasi

LiDAR *Light Detection and Ranger* o *Laser Imaging Detection and Ranging*

LIFT *Learn Invariant Feature Transform* [8]

LLBI *Latitude Longitude Bilinear Interpolation*

LSD-SLAM *Large-Scale Direct* monocular SLAM [9]

LSTM *Long short-term memory*

MLS *rigid Moving Least Squares*

MS-SSIM *Multi Scale Structural Similarity Index Method*

NNDR *Nearest Neighbor Distance Ratio*

OCamCalib *Omnidirectional Camera Calibration* [10]

ORB *Oriented FAST and rotated BRIEF* [11]

PAL *Panoramic Annular Lens*

PnP *Perspective* N-*Points*

RADAR *RAdio Detection And Ranging*

RANSAC *RANdom SAmple Consensus*

RCNN *Recurrent Convolutional Neural Networks*

RGB-D *Red Green Blue-Depth*

SfM *Structure-from-Motion*

SLAM *Simultaneous Localization And Mapping*

SONAR *SOund Navigation And Ranging*

SPHORB *Spherical* ORB [12]

SSIM *Structured Similarity Indexing Method* [13]

SURF *Speeded Up Robust Features* [14]

TILDE *Temporally Invariant Learned Detector* [15]

UCM *Unified Camera Model* [16]

1 Introducción

1.1 Motivación

La presente tesis se enmarca dentro del campo de la robótica móvil, cuyo impacto en la sociedad está en constante aumento. Las ventajas del uso de robots móviles autónomos e inteligentes son múltiples, sobre todo en ciertos ámbitos como en tareas de servicios, de asistencia o médicas. En [17], se presenta un repaso de algunos tipos de robots móviles empleados en la industria y en la sociedad. En esta misma línea, la figura 1.1 muestra algunos robots móviles comerciales. En primer lugar, hay robots móviles que se encuentran en nuestro día a día, como es el caso de los robots móviles para realizar servicios domésticos como limpieza, siendo uno de los más conocidos el robot aspirador Roomba, diseñado por iRobot [18]. En la sociedad cada vez se está recurriendo más a los robots móviles para realizar ciertas tareas, por ejemplo BellaBot fue diseñado por PUDU [19] para la entrega de alimentos. En cuanto a las industrias o fabricas, el robot RB-1 BASE fue diseñado por Robotnik [20] para transportar cargas en estos entornos. Por último, podemos encontrar a Pepper diseñado por Aldebaran [21] y que se suele emplear para asistencia de personas.

Los robots móviles autónomos pueden moverse de un lugar a otro para lograr sus objetivos deseados sin asistencia de agentes humanos [22], es decir, actúan sin comandos externos, usando únicamente la información obtenida a través de los sensores [23]. Para lograr esto, los robots móviles autónomos deben ser capaces de abordar, de forma precisa, determinadas tareas.

Durante la navegación autónoma, en todo momento, el robot debe conocer su posición (tarea de localización), conocer el entorno en el que actúa (tarea de creación de mapas), debería ser competente para la toma de decisiones respecto a la trayectoria

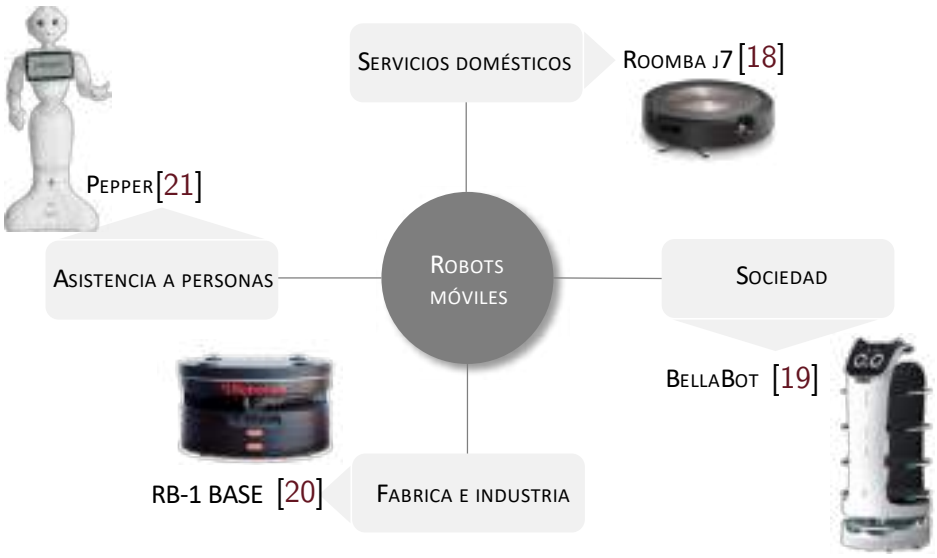


Figura 1.1: Se muestran cuatro ejemplos de robots móviles comerciales (RB-1 BASE [20], roomba j-7 [18], BellaBot [19] y Pepper [21]) y sus respectivas aplicaciones o ámbitos para las que fueron diseñados.

(tarea de planificación de trayectoria) y al movimiento (tarea de planificación de movimiento) y, por último, ser capaz de ejecutar el movimiento adecuado (tarea de control) [24]. Por su importancia y su complejidad, numerosos investigadores han centrado sus trabajos en las tareas de localización y creación de mapas.

Teniendo en cuenta que la localización se centra en estimar la posición y la orientación actual del robot móvil dentro del mapa del entorno, y que a menudo ese mapa es inicialmente desconocido y se debe ir creando a medida que el robot se mueve por el entorno, es muy habitual que las tareas de localización y de creación de mapas se resuelvan a la vez, de tal modo que al conjunto de estas dos tareas se le conoce como localización y mapeo simultáneos (SLAM, siglas del inglés *Simultaneous Localization And Mapping*). Así, el robot móvil crea el mapa del entorno a la vez que se localiza dentro de ese mapa.

Para poder resolver las tareas mencionadas, se requiere el uso de sensores que proporcionen información acerca del robot y/o del entorno. Por ejemplo, en lo que respecta a la tarea de localización, los más empleados en la robótica móvil son el sistema de posicionamiento global (GPS, del inglés *Global Positioning System*), los *encoders*, el LiDAR (acrónimo del inglés, *Light Detection and Ranger* o *Laser Imaging Detection and Ranging*) y los sistemas de visión. Los dos últimos sensores que se han mencionados son los utilizados frecuentemente para resolver el problema de SLAM. En [25], se puede encontrar más información sobre los sensores exteroceptivos: definición, aplicación, así como las ventajas e inconvenientes que presentan.

Por ejemplo, con el GPS se tiene una localización absoluta, pues la posición (latitud, longitud y altitud) se realiza con respecto al sistema de coordenadas de la Tierra. A diferencia de otras soluciones de localización que se analizarán en un momento posterior, con este sensor no se produce una acumulación de errores. No obstante,

la precisión de la posición puede verse afectada por otros factores. Por ejemplo, esta imprecisión puede deberse a los propios satélites, a errores durante la propagación de la señal o a interferencias en la señal del satélite [26]. Por ello, no es una solución válida en términos generales, dado que no funciona adecuadamente en entornos interiores, y también presenta problemas en entornos exteriores (por ejemplo, en calles estrechas con edificios altos).

En el caso de otros sensores a los que nos referiremos a continuación, la localización se resuelve mediante la odometría, que obtiene la posición relativa con respecto a otra anterior. En este caso, se suele asumir que la posición inicial es el origen y se calcula la posición con respecto a esta, es decir, en este caso la localización es relativa al punto de partida. La técnica empleada para ello depende del sensor que se esté empleando.

Uno de estos sensores son los *encoders*, los cuales determinan cuánto han girado las ruedas del robot móvil y dicha información, junto con el modelo de movimiento, se pueden emplear para estimar la posición y la orientación con incertidumbres asociadas. Entre los inconvenientes más importantes de este enfoque se encuentran los errores no sistemáticos de la odometría, que provocan que el desplazamiento calculado a partir de la odometría no se corresponda con el desplazamiento real del robot.

El tercer sensor que se ha enumerado es el LiDAR. A pesar de que se trata de uno de los sensores más empleados, también suele ser más costoso. Entre sus ventajas se encuentra que es capaz de proporcionar información de profundidad. Si bien es cierto que no se ve afectado por condiciones de iluminación, lo que resulta beneficioso para poder utilizarse tanto de día como de noche, la adquisición de datos con este sensor sí que se ve influida por ciertas condiciones climáticas como lluvia o niebla. Además, no son aptos para largos alcances o distancias y presentan ciertos problemas por reflectividad.

La odometría con LiDAR puede resolverse como un problema de registro de la nube de puntos actual y otra nube de puntos que es la de referencia. Li et al. [27] presentan varios métodos para el registro de nubes de puntos, clasificándolos en métodos basados en ICP (acrónimo del inglés *Iterative Closest Point*), métodos basados en características, métodos basados en aprendizaje, y métodos probabilísticos, entre otros.

Además de los anteriores, los sistemas de visión son otro tipo de sensores que se encuentran muy presentes en la robótica móvil. La principal ventaja de estos es la gran cantidad de información del entorno, que son capaces de proporcionar. Por ejemplo, con estos sensores, se puede tener información sobre el color y la textura del entorno, sin embargo este tipo de información no es proporcionada por los sensores mencionados anteriormente. Esta característica hace que se utilicen en varias aplicaciones como segmentación semántica o detección de objetos. Otras características que hacen que sean una buena selección para ciertas aplicaciones son su bajo coste y tamaño. Pese a todo ello, hay que tener en cuenta que se ven afectados por cambios en las condiciones de iluminación y climáticas. Además, no todas las configuraciones proporcionan información de profundidad.

En cuanto a las técnicas que se emplean con este tipo de sensor, la odometría visual se centra en resolver la localización como una estimación incremental del movimiento del robot a partir de las variaciones que dicho movimiento produce en las imágenes

que captura la cámara que se encuentra a bordo del mismo. Las técnicas de odometría visual se clasifican en dos grupos: métodos directos o basados en características. Mientras que los primeros se centran en minimizar los errores fotométricos, los segundos tienen como finalidad minimizar el error de reproyección con puntos característicos. Centrándonos en los métodos de odometría visual basada en características, existen diferentes procedimientos dependiendo fundamentalmente de la dimensión de las correspondencias a partir de las cuales se estimará el movimiento. Todo ello se encuentra detallado en [28, 29].

Los métodos basados en apariencia (o directos) utilizan toda la imagen, lo que hace que usualmente sean más lentos, pero por contra proporcionan una reconstrucción más densa del entorno ya que se utilizan todos los píxeles a diferencia de los métodos basados en características que únicamente hacen una reconstrucción en base a estas características. Sin embargo, la efectividad de los métodos basados en características se ve limitada en aquellos entornos sin o con poca textura. Centrándonos en esto último, se podría decir que los métodos directos son una buena solución en dichas condiciones, pero hay que tener en cuenta que no son adecuados para entornos de gran escala, es decir, solo son válidos para movimientos pequeños. Dado que ambos grupos se complementan en cuanto a ventajas e inconvenientes se refiere, existe un tercer grupo que son los métodos híbridos, que combinan los otros dos métodos. Estos métodos se basan tanto en el rastreo de características como en la utilización de la información de los píxeles de toda la imagen [30].

Cabe destacar que, cuando se trata de LiDAR o de un sistema de visión, además de obtener la posición mediante métodos incrementales, es decir, estimar la pose actual con respecto a una anterior como se ha visto en los párrafos anteriores, también se puede resolver la localización haciendo uso de un mapa que contiene puntos de referencia (u observaciones) cuyas posiciones son conocidas. De esta forma, se estima la posición con respecto a dichos puntos de referencia. Para ello, la posición de estos puntos de referencia debe ser lo suficientemente precisa pues, en caso contrario, puede producir un error inasumible la estimación de la pose. En este trabajo nos centraremos en el desarrollo de métodos de odometría visual, debido a las ventajas que presentan las cámaras frente al resto de sensores y a la importancia de este problema en el marco de los procesos de localización probabilística.

Los sistemas de visión se emplean con frecuencia en aplicaciones de robótica móvil como consecuencia de las ventajas que presentan, las cuales se han enumerado anteriormente. Sin embargo, una de las características de este tipo de sensor, que es el campo de visión, en ocasiones puede producir ciertas limitaciones en el ámbito de la robótica móvil. Por ejemplo, Suzuki y Suda [31] mencionan las limitaciones del campo de visión en detección de obstáculos para evitar colisiones. Aunque se centran en teleoperación de robots móviles, también podrían aplicarse a la navegación autónoma del robot. Entre las limitaciones que mencionan, se encuentra, por un lado, la pérdida de visión del objeto cuando el robot cambia de dirección (al igual que ocurre cuando el robot lo deja atrás), de tal forma que, para poder observarlo de nuevo, el robot debería parar y rotar. Además, consideran que un bajo campo de visión también es un inconveniente cuando el robot pasa por puertas o por caminos estrechos.

Esta limitación se puede resolver utilizando un tipo específico de sistema de visión,

el cual se conoce como omnidireccional. En comparación con las cámaras convencionales, las cámaras omnidireccionales cubren un mayor campo de visión del entorno alrededor del robot. Esto hace que el robot disponga de más información del entorno mientras se mueve, pues observa los objetos que se encuentran a su alrededor, no solo los que se encuentran delante del mismo. Además, en cuanto a localización y creación de mapas se refiere, con este sistema de visión se necesita de un menor número de imágenes para conocer el entorno. Así, en referencia a la localización, las características se seguirán observando por un mayor tiempo, puesto que se captura más información alrededor del robot, lo que conlleva una mayor solidez.

Los sistemas de visión omnidireccionales pueden presentar diferentes configuraciones, aunque las más empleadas en la robótica móvil son las cámaras catadióptricas, las cámaras con lente *fisheye* o múltiples cámaras. Por ejemplo, como consecuencia de las limitaciones de los entornos más complejos, Yang et al. [32] proponen un método SLAM de visión panorámica basado en la colaboración de múltiples cámaras, empleando el amplio campo de visión que proporcionan así como la región superpuesta entre las cámaras. En cuanto a los sistemas omnidireccionales monoculares, por un lado, Valiente et al. [33] presentan una serie de contribuciones con el objetivo de conseguir una técnica de localización robusta empleando una cámara catadióptrica. Por otro lado, Liu et al. [34] proponen un sistema de SLAM en tiempo real con una cámara con lente *fisheye* el cual es una extensión de ORB-SLAM [35, 36] introduciendo el modelo de cámara unificada mejorada.

Cada uno de los tres tipos de configuración presentan ventajas y desventajas. Por ejemplo, en el caso de las cámaras catadióptricas, en cada instante (o disparo) se tiene una única imagen con un amplio campo de visión. Esto es una ventaja, ya que requiere de un menor procesamiento y la imagen no presentará efectos debido a la unión de varias imágenes, lo cual suele dar lugar a una menor calidad de la imagen en las zonas de solape. En contra, se puede mencionar que este sistema de visión suele tener un tamaño mayor, lo que puede ser un inconveniente para ciertas aplicaciones. Asimismo, en cuanto a la imagen que proporciona, suele ser de menor resolución y aunque se puede conseguir un campo de visión de 360° en el eje horizontal, en el vertical dependerá de la geometría del espejo, pero en cualquier caso será menor a 180°. En el caso de tener una cámara con una lente *fisheye*, el campo de visión que se captura es menor que con una cámara catadióptrica (aproximadamente una semi esfera). Sin embargo, es posible capturar una esfera completa, es decir, un campo de visión de 360° en el eje horizontal y 180° en el vertical, posicionando dos lentes *fisheye* con sus respectivos sensores CMOS (siglas del inglés *Complementary Metal Oxide Semiconductor*) espalda contra espalda. Con esta configuración se solventa la limitación del campo de visión, pero conseguir una única imagen con todo el campo de visión no es una tarea sencilla, pues se necesita procesar el par de imágenes *fisheye* para poder fusionar la información capturada en cada una de ellas, lo que se conoce como técnica de *stitching*. Por ejemplo, Yao et al. [37] proponen un algoritmo que se compone de cuatro etapas (corrección de color, deformación, alineación y fusión) para la unión de imágenes capturadas por una cámara *fisheye* dual basado en puntos característicos. En la etapa de alineación, cada zona de solape, izquierda/derecha, se alinea por separado usando diferentes matrices de rotación. Para ello, se calcula la

Figura 1.2: A la izquierda, se muestra la cámara Garmin VIRB 360 y, a la derecha una vista completa proporcionada por la misma. En esta última se pueden apreciar efectos indeseados en la zona de solape, concretamente cuando hay información visual relevante.

diferencia de ángulos de n pares de puntos de correspondencias como el arcoseno de la diferencia de las coordenadas verticales $(y_2 - y_1)$, pues los autores determinan que en la dirección horizontal no difieren sustancialmente. Con el valor promedio de la diferencia de ángulo obtenido, la imagen se rota en la esfera.

El problema al unir varias imágenes en una sola es que si no se produce un correcto procesamiento (registro y *blending*) pueden aparecer ciertos efectos indeseados en las zonas de solape. Esto también ocurre en general con los sistemas de visión con múltiple cámaras. La principal diferencia entre utilizar múltiples cámaras o dos lentes *fisheye* opuestas es que, si bien es cierto que con las segundas se obtienen imágenes con mayor distorsión, lo que puede conllevar ciertos desafíos, solo con dos imágenes de este tipo se puede conseguir una vista completa del entorno. Así, para conseguir completar todo el campo de visión con la primera configuración, se necesitarían muchas imágenes y por lo tanto habrían más zonas en las que podrían aparecer los efectos indeseados de un incorrecto *stitching*.

En relación con lo anterior, conseguir una vista completa del entorno con una alta resolución y calidad a partir de un par de imágenes *fisheye* es otra línea importante de estudio, y es el segundo objetivo de la presente tesis. Como se observa en la figura 1.2, la cámara escogida para este trabajo (i.e. Garmin VIRB 360) puede proporcionar una vista completa directamente ya que realiza el proceso de *stitching* de forma interna. Sin embargo, se observa que en aquellas zonas ricas en textura, aparecen con frecuencia ciertos efectos indeseados que pueden implicar un problema para la navegación del robot. Por todo ello, se estableció esta segunda línea de investigación en el presente trabajo.

1.2 Objetivos

Los principales objetivos de esta tesis son, por un lado, resolver la tarea de localización, abordada como un problema de odometría visual, y, por otro lado, generar una vista completa con alta calidad del entorno, a partir de un par de imágenes *fisheye*.

Asimismo, será necesario disponer de una calibración robusta de la cámara utilizada, teniendo en cuenta las especificidades provocadas por el elevado campo de visión de las lentes. De este modo, en los siguientes capítulos se abordan estos tres grandes objetivos:

- **Calibrar la cámara Garmin VIRB 360 y realizar un estudio comparativo del desempeño de diversos algoritmos**. Teniendo presente que el proceso de calibración es importante en algunas aplicaciones de la robótica móvil, se sugiere:
  - Realizar un estudio de los métodos propuestos para este tipo de imágenes.
  - Estudiar algunos de ellos con esta cámara ya que sus lentes *fisheye* poseen un elevado campo de visión de $201.8°$ cada una, lo cual supone un problema para algunos modelos propuestos.
- **Resolver el problema de localización a partir de imágenes con un amplio campo de visión.** La tarea de localización en el ámbito de la robótica móvil es un aspecto clave para la realización de determinadas tareas. De modo que, el robot móvil debe localizarse con la mayor precisión posible, pues en caso contrario, pueden producirse comportamientos indeseados durante la realización de la tarea asignada. Para ello, se propone:
  - Estudiar un algoritmo de odometría visual basado en características con imágenes que contienen un gran campo de visión del entorno y que han sido obtenidas por dos configuraciones de sistema de visión diferente: (a) cámara con espejo y (b) cámara con lente *fisheye*.
  - Mejorar la búsqueda de correspondencias para que el movimiento relativo entre dos poses se obtenga con el menor número de falsos positivos. En otras palabras, conseguir una mayor robustez en esta parte del algoritmo.
  - Analizar la influencia de varios tipos de características locales en la estimación de la pose relativa con imágenes de gran angular, como son las capturadas por una cámara con lente *fisheye*.
- **Generar una vista completa del entorno de gran calidad, a partir de un par de imágenes capturadas por un sistema de visión con dos sensores, cada uno de ellos equipados con una lente *fisheye* con un campo de visión mayor a $180°$ y posicionados espalda contra espalda.** Para conseguir este objetivo principal, se propone:
  - Analizar la influencia, en la calidad de la vista completa, de la función que se utiliza para relacionar los puntos 3D y la imagen *fisheye*. Cabe tener en cuenta que para este tipo de sistema se han propuesto una gran cantidad de modelos.
  - Sabiendo que la parte de registro en el algoritmo tiene una gran peso en la calidad de la vista completa final y que las imágenes esféricas (o equirectangulares) no siguen una linealidad, se propondrá utilizar otro tipo de transformación geométrica durante esta parte del algoritmo que no sea lineal.
  - Aunque el alineamiento se lleve a cabo en la parte de registro, se estudiará

la relación entre la información visual común en el par de imágenes, que es lo que se va a fusionar, con el objetivo de minimizar esa diferencia de alineamiento antes de llegar a la parte de registro. De esta forma, el alineamiento de la información visual común se llevará a cabo con dos pasos.

- Proponer un método de evaluación de la calidad de la vista final automático.

1.3 Estructura

La presente tesis se encuentra organizada de la siguiente forma:

- Capítulo 2. En este capítulo se presentan los trabajos de investigación más relevantes relacionados con las líneas de investigación de esta tesis. En el apartado 2.1, se enumeran las principales tareas que un robot móvil debe resolver para realizar una navegación autónoma. En relación con esto, el apartado 2.2 describe algunos de los sensores más empleados en la robótica móvil, cuáles son sus principales ventajas e inconvenientes, así como trabajos en los que se emplean. Teniendo en cuenta que la presente tesis se centra en los sistemas de visión, el apartado 2.3 está dedicado a dicho tipo de sensor y en él se mencionan las diferentes configuraciones que se pueden encontrar, poniendo mayor énfasis en los sistemas de visión omnidireccionales. También se describen los diferentes procedimientos para adquirir una imagen con un amplio campo de visión del entorno. A su vez, el apartado 2.4 describe los dos métodos que existen para extraer la información visual de las imágenes, basados en características locales o globales, además de algunos de los algoritmo propuestos en la literatura para cada uno de ellos. Para finalizar, se presentan algunas de las técnicas propuestas para resolver la creación de mapas, en el apartado 2.5, y el problema de localización, en el apartado 2.6.
- Capítulo 3. Este capítulo se centra en el sistema de visión escogido para la presente tesis. Dado que se trata de un sistema de visión omnidireccional, el apartado 3.1 introduce este tipo de sensor además de mencionar algunos de los modelos de cámaras propuestos para dos de las configuraciones más empleadas, las cámaras equipadas con un espejo (catadióptricas) o con una lente *fisheye*, centrándose más en estas últimas. En el apartado 3.2 se describen las principales características de la cámara Garmin VIRB 360 que es el sistema de visión con campo de visión de 360° en el que se centra la presente tesis. Teniendo en cuenta que para varias aplicaciones, como la geometría epipolar o la rectificación de imágenes, se necesita conocer los parámetros intrínsecos de la cámara, en los dos siguientes apartados se estudia el proceso de calibración de la cámara mencionada. Por un lado, el apartado 3.3 describe la aplicación para calibrar cámaras con lente *fisheye* denominada Basalt [38] la cual tiene implementados varios modelos de cámaras. También se muestran los resultados obtenidos con cada modelo. Por otro lado, el apartado 3.4 describe otra herramienta para el mismo propósito que es OCamCalib [10] y también muestra los resultados obtenidos.
- Capítulo 4. Se introduce un enfoque probabilístico aplicado para la estimación de la pose relativa a partir de un par de imágenes. A dicho enfoque se le denomina *Adaptive Probability-Oriented Feature Matching* (APOFM) [1]. En el apartado

4.1, se introduce el tema que se va a tratar, se describe brevemente en qué consiste APOFM [1] y las contribuciones que se aportan. Los dos siguientes apartados describen dos modelos de estimación de movimiento, a partir de información visual y la matriz esencial en el apartado 4.2 y a partir del modelo de movimiento de odometría en el apartado 4.3. El apartado 4.4 describe, más detalladamente, todos los pasos del algoritmo de APOFM con las contribuciones implementadas. En el apartado 4.5, se muestran los resultados de los experimentos realizados. En primer lugar, con el objetivo de evaluar las aportaciones sugeridas y determinar si estas favorecen al método APOFM en cuanto a robustez en la búsqueda de correspondencias de puntos característicos así como en la solución a la estimación de la pose relativa. Dado que se pretende evaluar bajo estos dos aspectos, también se comparan los resultados, además de con el algoritmo anterior, con un método base para la estimación de la pose relativa con y sin implementar RANSAC para detectar las falsas correspondencias. En segundo lugar, se realiza otro experimento, únicamente con imágenes *fisheye*, para evaluar la influencia del método de extracción y descripción de características locales en este algoritmo y así escoger aquel que proporcione una estimación de la pose relativa más precisa.

- Capítulo 5. Este capítulo se centra en la generación de una vista completa del entorno con una cámara compuesta por dos lentes *fisheye* opuestas, como es la Garmin VIRB 360. Esta cámara puede configurarse para que proporcione una imagen de este tipo directamente. Pese a ello, se ha observado que cuando hay información visual rica en textura en la zona de solape, se pierde parte de información del entorno. Teniendo en cuenta que esto puede suponer un problema durante la navegación de un robot móvil con esta cámara a bordo, el objetivo de este capítulo es generar una vista completa con esta cámara donde este efecto se minimice lo máximo posible. De este modo, en el apartado 5.1, se realiza una breve introducción sobre este tipo de cámaras y se enumeran los principales desafíos que aparecen al generar la vista completa con las cámaras con doble lente *fisheye* opuestas. También se enumeran las diferentes propuestas. Después, el apartado 5.2 expone diferentes trabajos que se encuentran en la literatura para este mismo propósito. Asimismo, el apartado 5.3 está dedicado al algoritmo base para obtener la vista completa, describiendo en detalle los principales módulos proponiendo además diferentes alternativas en varios de los pasos. En referencia al primer módulo, hay dos alternativas cuya diferencia radica en la forma de proyectar sobre la esfera unitaria para transformar las imágenes *fisheye* a formato esférico. Este mapeo se puede realizar mediante un modelo de calibración (con los parámetros obtenidos en el capítulo 3) o un modelo de proyección que únicamente utiliza el campo de visión de la lente. El objetivo de este capítulo es evaluar la calidad de la vista completa resultante de ejecutar el algoritmo y compararla con la proporcionada por la cámara. Atendiendo a esto, el apartado 5.4 describe los métodos de medida de la calidad de la imagen resultante que se emplearán, donde uno de ellos es una contribución de esta tesis y consiste en un método de evaluación de la zona de solape empleando redes neuronales. Como se ha comentado, el objetivo de este capítulo es crear una vista completa con la mayor calidad posible. Por ello, el apartado 5.5 muestra los resultados de un experimento, cuya finalidad es conocer la relación entre los dos tipos de

imágenes con los que se trabaja en este algoritmo: par de imágenes *fisheye* y par de imágenes en formato esférico (equirectangulares). Así, tras analizar los resultados, se presentan dos contribuciones al algoritmo base. Por un lado, se propone incorporar un paso de corrección al primer módulo que relaciona el par de imágenes *fisheye*. Por otro lado, se propone utilizar, en el módulo de registro, un polinomio como transformación geométrica entre el par de imágenes equirectangulares. El apartado 5.6 describe el *dataset* utilizado en este capítulo. En el apartado 5.7, se describen los experimentos realizados y se muestran los resultados obtenidos. El apartado 5.8 describe la interfaz gráfica desarrollada en la que se encuentra implementado todo lo descrito en este capítulo.

- Capítulo 6. Se resumen las principales aportaciones y conclusiones, además de proponer posibles futuros trabajos derivados de las investigaciones realizadas en la presente tesis.

1.4 Resumen de materiales, métodos y discusión de resultados

1.4.1 Materiales

En este apartado se mencionan los materiales que se han empleado para poder llevar a cabo las investigaciones del presente trabajo:

- Como se ha mencionado, este trabajo se centra en el empleo de información visual proporcionada por un sistema de visión de 360°. En concreto, el sistema utilizado es la cámara Garmin VIRB 360, la cual se describe en el capítulo 3.
- Bases de datos de imágenes omnidireccionales, capturadas mediante una configuración de cámara *fisheye* y catadióptrica, de acceso abierto y disponibles en [39].
- Base de datos propia, compuesta por distintos tipos de imágenes adquiridas por la Garmin VIRB 360. Parte de esta base de datos se encuentra disponible en [40].
- PC con CPU Intel Core i7-10700 R a 2.90GHz

1.4.2 Métodos

- Herramientas de calibración de cámaras omnidireccionales: Basalt [38] y OCam-Calib [10] .
- Métodos de calibración de cámaras omnidireccionales: modelo de cámara unificado [16], modelo de cámara unificado ampliado [4], modelo de proyección de cámara doble esfera [3], Kannala-Brandt [7] y modelo de cámara basado en polinomio de Taylor propuesto por Scaramuzza et al. [10].
- Métodos de extracción de características locales: SURF (*Speeded Up Robust Features*) [14], ORB (*Oriented FAST and rotated BRIEF*) [11], FAST (*Features from Accelerated Segment Test*) [5], y KAZE [41].
- Herramientas matemáticas: Proceso Gaussiano [6], RANSAC (del ingés *RANdom SAmple Consensus*), geometría epipolar, matriz afín [42], método de los k vecinos más cercanos (en inglés, *k-nearest neighbors*, k-nn).

1.4.3 Resultados y discusión

A continuación, se resumen y exponen los resultados obtenidos en los diferentes experimentos que se han llevado a cabo en cada capítulo.

- Capítulo 3. Se muestran los resultados obtenidos al calibrar la cámara Garmin VIRB 360 con dos herramientas de acceso abierto: Basalt y OCamCalib. Cabe destacar que la primera herramienta se ha ejecutado con imágenes extraídas de dos vídeos, mientras que, para la segunda herramienta, se han adquirido directamente imágenes. Así que, la resolución de las imágenes no es la misma. Tras realizar una experimentación exhaustiva y un análisis comparativo de resultados, se llega a estas conclusiones:
  - En cuanto a los modelos implementados en Basalt, el propuesto por Kannala y Brandt [7] presenta un menor error de reproyección que los otros tres modelos ([16], [4] y [3]). El método que menor error de reproyección medio ha obtenido es el propuesto por Kannala y Brandt [7]. Aunque se ha observado que en muchos casos los puntos reproyectados no convergen cerca de las esquinas detectadas. Esto ocurre cuando el patrón se encuentra en la zona de mayor distorsión. De este modo, aunque el error es pequeño, este modelo no se ajusta correctamente a este tipo de cámara. Entre el resto de modelos, el menor error se ha conseguido con el propuesto por Khomutenko et al. [4].
- Capítulo 4. En este capítulo se exponen varias contribuciones al método APOFM (siglas del inglés *Adaptive Probability-Oriented Feature Matching*) [1] para estimar la pose relativa entre pares de imágenes de gran angular. Este enfoque construye un modelo del entorno con información de repetibilidad y el carácter distintivo de los puntos de la escena. Concretamente, este modelo es una distribución de probabilidad generada por un proceso gaussiano.
  - En el primer experimento se estudia la influencia de dos de los parámetros configurables del proceso gaussiano sobre dos aspectos claves en la localización, como son el tiempo de cálculo y el error cometido. Con este experimento, se seleccionan los valores óptimos para conseguir un balance entre los dos aspectos estudiados.
  - En el siguiente experimento se evalúan las contribuciones propuestas sobre el algoritmo de APOFM. Para poder establecer si dichas contribuciones mejoran el algoritmo base [1], se analizan los resultados sobre el número de correspondencias de puntos característicos y falsos positivos (ya que fue propuesto para una búsqueda de correspondencia más robusta), así como el error de localización (pues se aplica en un algoritmo para la estimación de la pose relativa). En este experimento, con objeto de evaluar el método APOFM frente a otros algoritmos, también se ejecuta un algoritmo base de estimación de la pose relativa e implementando RANSAC, para comparar los resultados. Tras estudiar los resultados, el hecho de utilizar una búsqueda de correspondencias ponderada a partir de la información de probabilidad del modelo, como se ha propuesto, mejora la robustez del algoritmo.
  - En el tercer experimento, se estudia la influencia del tipo de característica

local en la estimación de la pose relativa con un algoritmo base para ello e implementando el enfoque APOFM. Los resultados determinan que ORB [11] proporciona un menor error de estimación de la pose relativa y que KAZE [41] funciona mejor con el algoritmo base que empleando APOFM.

- Capítulo 5. En este capítulo se genera una vista completa del entorno a partir de un par de imágenes *fisheye*. El algoritmo para ello tiene implementados varios procedimientos para un mismo paso, de modo que hay diferentes propuestas para obtener la vista completa. Por un lado, en la primera etapa, se han implementado dos opciones para el mapeo entre imagen *fisheye* y esfera: el modelo de cámara propuesto por Scaramuzza et al. [10] o la proyección equidistante. Además, en esta etapa, también se ha implementado un paso de corrección, propuesto en esta tesis doctoral, que está diseñado para ser utilizado junto con la proyección equidistante. De este modo, para esta primera etapa hay cuatro variantes: (a) modelo de cámara propuesto por Scaramuzza et al. [10], (b) proyección equidistante (sin aplicar paso de corrección propuesto), (c) proyección equidistante con corrección de una de las coordenadas y (d) proyección equidistante con corrección de las dos coordenadas. Por otro lado, en la segunda etapa, se han implementado dos opciones de transformación geométrica: (I) mediante la matriz afín y (II) haciendo uso de un polinomio, propuesto también en esta tesis doctoral.

  - En el primer experimento, se evalúan las zonas de solape de las vistas completas generadas con las dos primeras variantes de la primera etapa, (a) y (b), además de la vista proporcionada directamente por la Garmin VIRB 360. Para este experimento, se utiliza el método de evaluación propuesto que se basa en la obtención de una medida de distancia entre descriptores extraídos de una de las capas de una red neuronal. También se realiza una evaluación que consiste en determinar cuántas marcas ArUco es posible identificar. De estos resultados se obtiene que la mejor calidad se consigue al realizar el mapeo utilizando el modelo propuesto por Scaramuzza et al. [10] con sus parámetros intrínsecos.

  - En el segundo experimento, se evalúan las contribuciones propuestas con respecto al algoritmo, como son el paso de corrección y la utilización de un polinomio para alinear el par de imágenes equirectangulares. En primer lugar, se analiza y compara la calidad de cinco tipos de vistas completas con una medida que se basa en la nitidez y que no requiere de una referencia. Estas cinco vistas corresponden a la proporcionada por la Garmin VIRB 360 y las generadas al escoger las opciones (a), (b), (c) y (d) con la matriz afín. Con esta medida, la peor calidad, en la mayoría de los casos, pertenece a la vista proporcionada por la Garmin VIRB 360 y la mejor a la generada con el modelo propuesto por Scaramuzza et al. [10]. En segundo lugar, se emplea una medida de calidad de la imagen con referencia, como es MS-SSIM, para evaluar las cuatro variaciones de la primera etapa del algoritmo, (a), (b), (c) y (d), utilizando además la matriz afín y el polinomio en la segunda fase, comparando así todas las posibles variaciones del algoritmo completo. Aparte de la medida de calidad, también se estudian otros aspectos de estas vistas como son el tiempo computacional y la diferencia entre los

pares de correspondencias de las imágenes equirectangulares. De todo ello, los resultados determinan que el paso de corrección, concretamente el de las dos coordenadas, hace que la calidad de la vista completa en la zona de solape mejore, con respecto a no utilizar el paso de corrección, es decir, (b), e incluso igual a la calidad obtenida al utilizar el modelo propuesto por Scaramuzza et al. [10], pero sin necesidad de calibrar el sistema de visión. En cuanto a la transformación geométrica, los resultados son mejores al utilizar el polinomio que la matriz afín.

2 Estado del arte

El hecho de que los robots móviles puedan moverse y que no se encuentren anclados como los robots manipuladores ha permitido que puedan aplicarse en diversos ámbitos como, por ejemplo, en la industria, en las instituciones, en la agricultura [43], en los hogares [44], etc.

En este sentido, a lo largo de los años los robots móviles se han empleado para llevar a cabo ciertas tareas que anteriormente realizaban los seres humanos pero que, o bien dicha actividad, o bien el entorno en el que debe ejecutarse, suponen algún peligro para estos (tales como misiones de rescate [45], misiones en el espacio [46], en minas [47], etc). Más recientemente, su campo de aplicación se ha visto enormemente ampliado, de tal forma que en la actualidad podemos encontrar el uso de los mismos como complemento en determinados ámbitos profesionales, como en la entrega de paquetes [48, 49], como apoyo a la medicina [50], en instituciones académicas como apoyo a la docencia [51], en restaurantes [52-54] o en aeropuertos [55, 56]. También se suele recurrir a ellos para mejorar servicios y actividades cotidianas, del día a día, como tareas de limpieza en hogares [57] o asistiendo a personas [58, 59], entre otras.

A este respecto, podemos decir que existen cuatro campos tecnológicos cuyo desarrollo es importante para conseguir la autonomía de los robots móviles: locomoción, percepción, cognición y navegación [60]. Todos estos aspectos se estudian en [61] para robots móviles en entornos interiores.

2.1 Navegación autónoma

Muchas de las aplicaciones mencionadas requieren que el robot móvil navegue por un entorno impredecible, dinámico y a priori desconocido, tomando él mismo (sin la

asistencia del ser humano) las decisiones oportunas para desempeñar de forma eficaz la tarea para la que ha sido diseñado.

En este sentido, para una navegación autónoma y segura, se plantean diversos retos y problemáticas a las que debe ofrecerse una solución adecuada (ver figura 2.1): ha de localizarse correctamente en el entorno y ser capaz de crear un mapa del mismo de forma simultánea (SLAM [62, 63], del inglés *Simultaneous Localization and Mapping*), sobre el que tiene que planificar la trayectoria [64, 65] óptima que realizará, teniendo en cuenta los posibles obstáculos para evitarlos.

Figura 2.1: Desafíos durante la navegación autónoma.

Como se puede comprobar, la localización es una tarea relevante en la navegación autónoma, ya que, si el robot no es capaz de saber su posición y orientación en todo momento, no podrá estimar la trayectoria para llegar a su destino ni tampoco calcular su posición con respecto a otros objetos para detectar posibles colisiones.

Cabe señalar que un aspecto clave para realizar una tarea de forma autónoma es la adquisición de información, ya sea interna del propio robot móvil (sensores propioceptivos), o ya sea del entorno (sensores exteroceptivos). Como se comentará en el apartado 2.2, existe una gran variedad de sensores que se pueden incorporar en el robot móvil. La elección del tipo de sensor o sensores dependerá del entorno en el que vaya a realizar la tarea (terrestre, aéreo, o acuático [66]), del diseño del mismo (robots móviles con ruedas, con patas o aéreos) y de la aplicación.

En la actualidad, el interés por el uso de visión para resolver la tarea de localización y creación de mapas ha aumentado. Chen et al. [67] recogen, en una gráfica, el número de citas a diferentes métodos para SLAM (en función del sensor o sensores que emplean) desde 2003 hasta 2022 en la Web of Science. En dicha gráfica se puede ver que los que emplean visión tienen un mayor impacto (más citas) en los últimos años con respecto a otros como el LiDAR (*Light Detection and Ranger* o *Laser Imaging Detection and Ranging*) que es el segundo.

El algoritmo correspondiente a Visual-SLAM se estructura generalmente en dos bloques: *frontend* y *backend* [68]. El primero de ellos corresponde a la parte en la que se realiza una estimación local de la trayectoria de la cámara (mediante odometría visual). Esta información se envía al segundo bloque, *backend*, donde se optimiza el mapa creado. Aparte de estos dos bloques, algunos algoritmos suelen añadir alguno más como el cierre de bucle, cuya finalidad es reconocer las ubicaciones que han sido visitadas previamente, consiguiendo de esta forma un mapa más preciso.

En el *frontend*, a medida que el robot navega por el entorno, se van capturando imágenes que son la entrada a un algoritmo de odometría que calcula, a partir de dos imágenes consecutivas, el movimiento de la cámara relativo entre esos dos instantes de tiempo. Sin embargo, los errores que se producen en la estimación entre un par de imágenes se van acumulando a medida que se realizan más estimaciones. Por lo tanto, tanto el *backend* como el cierre de bucle se emplean para disminuir esta acumulación de error.

Atendiendo a lo comentado en el párrafo anterior, la principal distinción entre Visual-SLAM y la odometría visual radica en si se está considerando o no la consistencia global del mapa, así como la trayectoria que se ha predicho [69].

Debido al gran avance y desarrollo de la inteligencia artificial, lo cierto es que, cada vez con más frecuencia, se pueden encontrar trabajos en los que se abordan uno o más módulos de Visual-SLAM con métodos de aprendizaje profundo. De hecho, Zhang et al. [70] recogen varios trabajos en los que integran esta técnica en odometría visual, detección de cierre de bucle y creación de mapas.

El campo de investigación en la navegación autónoma de los robots móviles se encuentra todavía en desarrollo, pues a pesar de que sean muchos los trabajos científicos relacionados, se sigue trabajando en mejorar las técnicas ya existentes. De hecho, una línea reciente es la incorporación de información de alto nivel (información semántica) en la navegación. Este tipo de información engloba objetos, lugares, así como la relación que existe entre ellos [71]. Chen et al. [72] muestran un estudio sobre los métodos propuestos para resolver la localización y creación de mapas simultaneo con visión. Estos métodos los divide en técnicas tradicionales y técnicas con información semántica basada en aprendizaje profundo.

2.2 Sensores en robótica móvil

Un sensor es un dispositivo sensible a ciertas magnitudes físicas o estímulos externos del entorno, que produce una señal que puede ser medida y procesada tanto de forma digital, en su gran mayoría de casos, como de forma analógica, en casos más tradicionales. En la robótica móvil, los sensores permiten que los robots adquieran información de su entorno, siendo necesarios para una navegación e interacción segura y efectiva [73].

A la hora de seleccionar un sensor u otro para incorporarlo en un robot, hemos de tener en cuenta que es importante conocer el entorno en el que el mismo va a navegar, así como la tarea asignada. A este respecto, no todos los sensores son aptos para cualquier tipo de actividad y entorno y resulta esencial escoger el que más se ajusta a

los intereses propuestos. Por ejemplo, para llevar a cabo las expediciones por parte de los robots de la NASA en otros planetas, no se puede utilizar el GPS (acrónimo del inglés: *Global Positioning System*) al no estar disponible fuera del planeta Tierra [74]. Otro aspecto que se ha de tener presente es la locomoción del robot móvil, ya que, por ejemplo, los *encoders* solo pueden aplicarse en robots móviles que tengan ruedas.

Uno de los sistemas más conocidos es el sistema de posicionamiento global (GPS), que es un tipo concreto de sistema global de navegación por satélite (GNSS, acrónimo del inglés: *Global Navigation Satellite System*). Disponer de este tipo de sensores es la forma más directa de conocer la posición. En cambio, no es un método adecuado cuando la tarea se lleva a cabo en entornos interiores. De hecho, aún en un entorno exterior, la precisión de este sensor puede verse disminuida en algunos momentos ante la presencia de ciertos elementos, como edificios, árboles o túneles, que bloquean la señal. En este sentido, muchos investigadores han estudiado y propuesto diferentes técnicas de localización para entornos en los que no se tiene acceso a GPS [75, 76]. Por ejemplo, Aftatah et al. [77] proponen una fusión de sensores para que un vehículo que navega por un entorno urbano pueda localizarse en todo momento. La solución de la localización se realiza fusionando las medidas de este sensor con las proporcionadas por un Sistema de Navegación Inercial (INS, acrónimo del inglés *Inertial Navigation System*) cuando la señal del GPS está disponible. Cuando esto último no ocurre, resuelven la localización con dos sensores propioceptivos que reemplazan el GPS por las medidas del odómetro.

Además de estos factores que pueden ocasionar imprecisiones, hay que tener en cuenta que la información que proporciona este sensor únicamente se puede emplear para conocer la localización absoluta, no proporciona otra información acerca del entorno. Por dicho motivo, se suele equipar al robot tanto con este último como con otros sensores. Patoliya et al. [78] describen un sistema de navegación en un entorno sin restricciones en el que integran, principalmente, los datos de GPS y LiDAR (acrónimo del inglés, *Light Detection and Ranger* o *Laser Imaging Detection and Ranging*). Para el movimiento autónomo del robot, se utilizan los datos proporcionados por GPS, mientras que los datos del LiDAR permiten identificar los posibles obstáculos existentes. Este último sensor se combina con los datos de la odometría e IMU para la generación del modelo del entorno empleando el algoritmo *Gmapping* basado en un filtro de partículas *Rao-Blackwellized* [79].

Al margen de los dispositivos GPS, podemos destacar otros sensores extensamente usados en robótica móvil, como los *encoders* (para robots con ruedas), unidades de medición inercial o IMU (siglas del inglés *Inertial Measurement Units*), SONAR (acrónimo del inglés *SOund Navigation And Ranging*), LiDAR y sistemas de visión. En la tabla 2.1, se muestra una comparación de todos los sensores mencionados en este párrafo con respecto al tipo de entorno terrestre en el que se pueden utilizar (exterior e interior), tarea que se puede resolver (localización y detección e identificación de obstáculos), tipo de información del entorno que proporciona (textura, color y profundidad) y si la información que proporciona es sensible ante ciertas condiciones (iluminación y clima).

En cuanto a los *encoders*, estos sensores se colocan en el extremo del eje del motor y convierten la posición lineal/angular de este en una señal ya sea digital o analógica. La técnica de localización empleando la información proporcionada por este tipo de sensor es conocida como odometría, término que dio origen a la odometría visual presentada

Tabla 2.1: Comparativa de los sensores más empleados en la robótica móvil. E = Exterior e I = Interior.

Sensor		GPS	*encoders*	IMU	LiDAR	SONAR	RADAR	Sistema de visión
Entorno		E	E/I	E/I	E/I	E/I	E/I	E/I
Tarea	Localización	✓	✓	✓	✓	✓	✓	✓
	Obstáculos	-	-	-	✓	✓	✓	✓
Tipo de inf. del entorno	Textura	-	-	-	✓	-	-	✓
	Color	-	-	-	-	-	-	✓
	Profundidad	-	-	-	✓	✓	✓	*
Sensibilidad	Iluminación	-	-	-	-	-	-	✓
	Condiciones climáticas	-	-	-	✓	✓	✓	✓

* Solo si es una cámara RGB-D o un sistema compuesto por más de una cámara con campo de visión común.

anteriormente. En cuanto a la precisión, cuando se emplean los *encoders* hay que tener en cuenta que el error es acumulativo, es decir, aumenta cuanto mayor es el tiempo que lleva el robot en movimiento. Las causas de dichos errores en la odometría se dividen en errores sistemáticos (como consecuencia de que el diámetro de las ruedas no sean completamente iguales o que no se encuentren correctamente alineadas, entre otros) y errores no sistemáticos (debido a condiciones externas: deslizamiento de las ruedas, suelo irregular, etc.) [80]. Por ejemplo, Onyekpe et al. [81] proponen WhONet, una red neuronal recurrente que aprende las incertidumbres que se producen al estimar el desplazamiento a partir de las mediciones sobre la velocidad de las ruedas. La finalidad de esta red es conseguir un posicionamiento más preciso. Por ello, los autores evalúan esta propuesta en varios entornos en los que hay presente alguna condición desafiante (como curvas cerradas o carreteras mojadas) que hace que la localización sea una tarea difícil. En comparación con esta estimación directa a partir de la señal de los *encoders* y de la cinemática inversa del robot, los resultados son mejores ante esas condiciones desafiantes, así como en aquellos entornos en los que se produce interrupción en la medida del GNSS.

Otro tipo de sensor empleado con frecuencia son las IMU. Formadas habitualmente por una combinación de giroscopios y acelerómetros. La información que se puede conseguir con este sensor es la posición, orientación, velocidad y aceleración con respecto a un sistema de referencia inercial. Inicialmente, se trataba de sensores de grandes di-

mensiones y muy complejos, lo cual dificultaba su incorporación en los robots móviles. No obstante, en la actualidad se han mejorado y encontramos IMU con tamaños muy reducidos, sofisticados y eficientes [82]. La desventaja que presenta este sensor es que, a medida que el robot se mueve, el error de posicionamiento será mayor, puesto que es acumulativo, a menos que se corrija, al igual que ocurre con el proceso de odometría. Por esto último, se suele combinar con otros sensores.

Los sensores descritos hasta el momento se emplean para resolver el problema de posicionamiento. Ahora nos centraremos en aquellos otros que dan al robot la capacidad de observar su entorno, es decir, en los sensores exteroceptivos como SONAR, RADAR, LiDAR o sistemas de visión.

El sensor LiDAR suele emplear fuentes de luz coherentes (láser), que se utilizan para medir la distancia, así como las propiedades reflectantes del entorno [73]. Se usa para resolver diversas tareas como la detección y reconocimiento de objetos [83-89], así como para la localización y creación del mapas [90-94]. En esta línea, Cattaneo et al. [95] proponen una red llamada LCDNet que detecta cierres de bucle a partir de nubes de puntos obtenidas con LiDAR. Para ello, se realiza, al mismo tiempo, una identificación de los lugares visitados y una estimación de la transformación relativa con seis grados de libertad entre el escaneo actual y el mapa. LCDNet se compone principalmente de una primera parte en la que se extraen características de los puntos, una segunda parte en la que se extraen descriptores globales y una tercera parte en la que se estima la pose relativa entre dos nubes de puntos. Liu et al. [96] ofrecen un estudio en el que, a partir de una nube de puntos 3D obtenida con LiDAR, se genera una proyección ortográfica (imagen a vista de pájaro). Después de ello, se extraen puntos clave y sus respectivos descriptores, a través de una red neuronal R2D2. Finalmente, con estos descriptores, se realiza una estimación de la pose empleando un procedimiento compuesto por dos fases.

La principal ventaja de este tipo de sensor en el ámbito de la robótica es que proporciona información de profundidad directamente. Además, son insensibles a los entornos de baja textura. Otro aspecto a considerar es que la lectura de este sensor no se ve afectada por la iluminación ambiental, pero sí por ciertos fenómenos meteorológicos (nieve, lluvia o niebla). Jokela et al. [97] prueban el funcionamiento de varios modelos de LiDAR en condiciones adversas como niebla y nieve. Del estudio realizado, concluyen que, como era de esperar, estas condiciones afectan a la información capturada.

Por otro lado, encontramos el SONAR, que es un dispositivo que se utiliza para detectar objetos y medir distancias mediante el empleo de ondas sonoras. A este respecto, encontramos dos tipos de SONAR: los SONAR pasivos, que solamente perciben sonidos; y los SONAR activos (generalmente denominados sensores ultrasónicos), que además emiten impulsos de sonido. En este último caso, este tipo se suele usar tanto para detectar y evitar obstáculos como para las tareas de localización [73].

En cuanto al RADAR (acrónimo del inglés *RAdio Detection And Ranging*), tal y como hemos visto con los LiDAR, se trata de dispositivos que emiten ondas electromagnéticas. Sin embargo, la diferencia radica en que en el RADAR se utilizan ondas de radio en lugar de luz láser. De esta forma, se emite y se recibe la señal por dos antenas diferentes (o antenas matrices) para estimar la posición de objetos [73]. Cabe

destacar que, a diferencia de otros sensores, los RADAR son más robustos y tienen una mayor resistencia frente a entornos y condiciones ambientales adversas. En todo caso, presentan una menor precisión al estar expuestos a posibles problemas de interferencia con el entorno, que puede ser dinámico y ruidoso. En esta línea, Hong et al. [98] proponen un sistema SLAM gráfico basado en este sensor (RadarSLAM). También evalúan la precisión de la localización de este sistema bajo distintas condiciones meteorológicas adversas. De estos estudios, se obtiene que la estimación de la pose durante una trayectoria se aproxima mucho al *ground truth* en la oscuridad y nieve. En el caso de la lluvia y la niebla, el error aumenta con el tiempo. Respecto a esto último, es importante señalar que las trayectorias seguidas para oscuridad/nieve y lluvia/niebla no son las mismas. En el caso de lluvia y niebla, no hay bucle. Atendiendo a esto, el aumento de error puede estar asociado a esta característica de la trayectoria y no a las condiciones ambientales.

En último lugar, encontramos los sistemas de visión (cámaras simples o cámaras RGB-D) que, a diferencia de los anteriores (en general, sistemas activos), son pasivos, esto es, son sensores que no necesariamente emiten energía al entorno. Debido a que es el sensor que se ha utilizado en la presente tesis, serán estudiados con más detalle en el apartado 2.3.

Los sensores que son capaces de proporcionar información de profundidad son el LiDAR, el SONAR y el RADAR, así como las cámaras RGB-D. Sin embargo, presentan ciertas limitaciones. Como consecuencia, y aprovechando el avance en la técnica de aprendizaje profundo, se han propuesto varios métodos basados en esta técnica para recuperar a partir de una imagen RGB el mapa de profundidad. Varios de estos métodos basados en aprendizaje profundo se encuentran descritos en [99, 100]

En la robótica móvil, los sensores que más se emplean son los sistemas de visión y los LiDAR debido a las propiedades que presentan. Además, se pueden emplear ambos de forma conjunta para aprovechar las principales ventajas de cada uno [101]. Por ejemplo, Chen et al. [102] combinan imagen RGB y LiDAR para mejorar las estrategias de fusión en la precisión de detección de vehículos 3D en múltiples condiciones de iluminación. En concreto, se lleva a cabo la combinación anterior porque la información proporcionada por el LiDAR que captura una nube de puntos 3D del entorno, permite codificar su forma explícitamente. Se produce, sin embargo, de forma dispersa y no garantiza una información detallada en cuanto a la textura. Por ello, se propone la incorporación de imagen RGB que ofrece una información ampliada de la textura y del color, aunque sin profundidad. Así las cosas, los dos sensores se complementan a la perfección y permiten, como apuntan los autores, aumentar la solidez de los resultados para la detección de objetos en 3D. En la misma línea, Silva et al. [103] fusionan las salidas de un escáner LiDAR y una cámara con gran angular. Esta fusión de los datos proporciona una mayor efectividad de los resultados en un algoritmo para la detección de espacio libre, es decir, sin obstáculos.

2.3 Sistemas de visión

Los sistemas de visión están adquiriendo en los últimos años mayor protagonismo dentro de la robótica, como consecuencia de las ventajas que presentan. En primer

lugar, la característica más interesante que posee este tipo de sensor es la cantidad de información del entorno que proporciona en una única captura. Además, dicha información es similar a la que nosotros percibimos de nuestro alrededor: iluminación, forma, textura, color, etc. Como consecuencia de esto, las cámaras permiten implementar aplicaciones adicionales a la navegación del robot, tal como la detección e identificación de objetos [104-106] o la extracción de información de alto nivel (semántica) [107-110]. Por último, también es importante mencionar otras características que son relevantes para ciertas aplicaciones, como su bajo coste, peso y tamaño. En la figura 2.2 se muestra, de forma esquemática, las principales ventajas y desventajas de estos sensores.

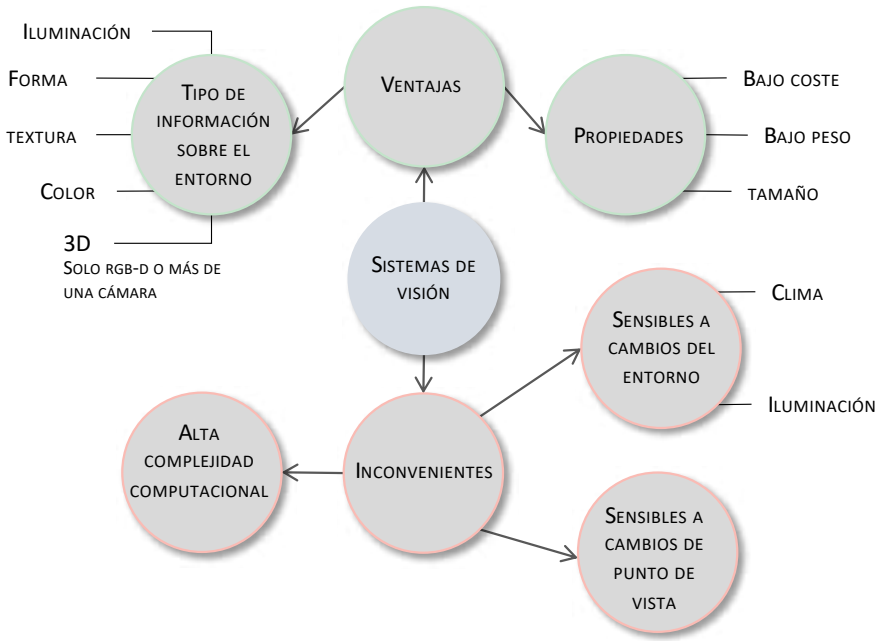


Figura 2.2: Ventajas e inconvenientes presentes en los sistemas de visión.

Por otro lado, entre las desventajas de este sensor, se encuentran las siguientes problemáticas. En primer lugar, si bien es cierto que es una ventaja que proporcione abundante información del entorno, esto conlleva que su procesamiento pueda requerir de una alta complejidad computacional, lo cual puede ser un impedimento en la práctica si se quiere realizar en tiempo real. En segundo lugar, los métodos basados en imágenes son sensibles a los factores externos, como cambios en el entorno (tiempo, estación o variación de la iluminación) y también a cambios de los puntos de vista. Por ello, se busca que sean invariantes ante estos cambios, lo cual es objeto de estudio en muchos trabajos de investigación. En tercer lugar, siguiendo con las limitaciones comentadas, una iluminación baja en el entorno también dificultará tareas como reconocimiento de objetos o localización, puesto que las características visuales serán difíciles de reconocer. Por ejemplo, esto se puede ver en los resultados de uno de los experimentos realizados en [111]. En dicho experimento los autores analizan la precisión de la detección de rostros en imágenes con poca iluminación. Para ello, comparan los resultados sobre las imágenes originales y estas mismas tras mejorarlas con varios algoritmos basados en *deep learning* para dicho propósito. La conclusión de dicho experimento es que el

número de detecciones erróneas es menor tras la mejora. Por el contrario, el número de detección correctas aumenta.

Siguiendo con los problemas que pueden suponer las condiciones de iluminación, Aladem et al. [112] centran su trabajo en la posibilidad de que el robot realice la navegación por la noche o con poca luz. Por ello, estudian cuatro técnicas de preprocesamiento de imágenes con la finalidad de fortalecer las tareas de odometría visual con características y SLAM visual en dichas condiciones. Además, también comparan los resultados de este estudio con los obtenidos empleando imágenes tomadas de día, obteniendo, como era de esperar, que en este caso se consigue la mejor solución.

Cebollada et al. [113] resuelven el problema de localización jerárquica, abordándolo como un reconocimiento de lugares basado en imágenes panorámicas, de un robot móvil que navega por un entorno de interior. Estudian varios métodos de extracción de descriptores, todos ellos de apariencia global, ante distintos cambios de iluminación.

2.3.1 Configuraciones de los sistemas de visión

Los sistemas de visión se pueden clasificar en función del número de cámaras empleadas: monoculares (una sola cámara, figura 2.3(a)), estéreo (dos cámaras, figura 2.3(b)) o múltiples cámaras (figura 2.3(c)).

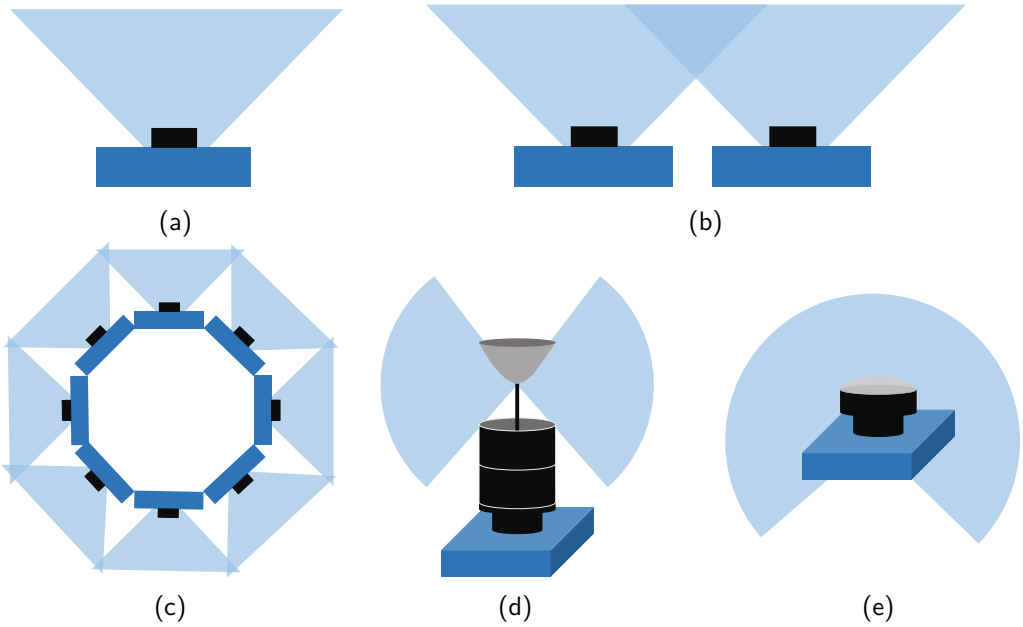


Figura 2.3: Configuraciones de los sistemas de visión: (a) monocular, (b) estéreo, (c) múltiples cámaras, (d) catadióptrica y (e) *fisheye*.

Para la navegación autónoma, podemos encontrar tanto trabajos en los que emplean un sistema monocular [35, 114-116] como un sistema de visión estéreo [117-120]. Por ejemplo, Badrloo et al. [121] describen varios métodos de detección de obstáculos a partir de imágenes y los dividen en: técnicas de detección de obstáculos monoculares o basados en pares estéreo.

El sistema de visión compuesto por una única cámara posee un menor tamaño y

peso que el compuesto por más de una de ellas. Por dicho motivo, se suele incorporar en robots de pequeño tamaño. Por ejemplo, es habitual equipar los vehículos aéreos no tripulados (UAV) pequeños con una cámara monocular para resolver las tareas de localización [122] o detección de objetos y evitar obstáculos [123].

Sin embargo, no es posible recuperar información de profundidad con solo una imagen. Por lo tanto, si lo que se necesita es obtener información 3D del entorno en un mismo instante, esto se consigue teniendo más de una cámara. Por ejemplo, Li et al. [110] emplean sistemas de visión estéreo para construir un mapa denso de puntos 3D a partir de un sistema Stereo-ORB-SLAM.

A pesar de que un sistema estéreo sea capaz de recuperar las coordenadas 3D mediante triangulación, es importante destacar que la precisión durante este procesamiento puede ser relativamente baja en ocasiones. Este hecho puede ser a consecuencia de que la distancia entre los centros de ambas cámaras se vea afectada por una imprecisión durante su medida o debido a una expansión térmica o bien contracción por frío (cambios climáticos) [124]. Cabe recordar que, como ya se ha comentado en el apartado 2.2, se han propuesto varias técnicas para recuperar información de profundidad a partir de una imagen. Otro aspecto a tener en cuenta al escoger un sistema u otro es que el proceso de calibración será más complejo cuando haya más de una cámara, pues será necesario calibrar cada una de ellas, así como conocer la relación 3D (rotación y translación relativas) entre ellas.

Una propiedad muy relevante, sobre todo cuando se emplean como sensores en robótica, es el campo de visión, es decir, la cantidad de información que es capaz de capturar una cámara del entorno. En sistemas compuestos por una única cámara, el campo de visión se puede aumentar con diferentes configuraciones, siendo posible capturar 360° alrededor del sistema de visión, lo que se conoce como sistema omnidireccional [125]. Siguiendo con esta misma categoría, el campo de visión se puede aumentar rotando dicha cámara o combinándola con una lente de gran angular (figura 2.3(e)) o con un espejo (figura 2.3(d)). En el caso de sistemas multicámara, otra forma de aumentar el campo de visión es posicionar varias cámaras hacia distintas direcciones pero de forma que exista zona de solape entre ellas. En resumen, las cámaras omnidireccionales se pueden dividir en configuraciones con una única cámara o con una matriz de cámaras, que se conoce como cámaras polidióptricas [126].

En relación con el uso de estas configuraciones en robótica móvil, las tres últimas configuraciones son factibles para obtener información del entorno durante la navegación, a diferencia de la primera que requeriría que, por un lado, el entorno sea estático y que, por otro lado, el robot se detenga en cada instante para que la cámara rote y obtenga así una vista de 360°.

Tener un mayor campo de visión proporciona ciertas ventajas. Por ejemplo, la diferencia de información visual entre un par de imágenes tomadas con cierta distancia será menor si el campo de visión es más amplio, por lo que, el número de características coincidentes aumentará. Otra ventaja es que una percepción de 360° alrededor del robot hace que la navegación por un entorno dinámico sea más segura ya que sería capaz de detectar personas u objetos que no se encuentran en la misma dirección de movimiento, sino que se encuentran en su alrededor pero que podrían ocasionar algún inconveniente

a la navegación (debido a la interacción de objetos no estáticos como personas o animales). Asimismo, se puede construir un modelo del entorno con un menor número de imágenes. No obstante, cabe plantear, como desventaja, que disponer de un amplio campo de visión conlleva habitualmente una menor resolución y, además, una mayor distorsión. Chappellet et al. [127] comparan el rendimiento de varios tipos de cámaras para resolver la tarea de localización y mapeo simultáneo. Las cámaras en las que se basa este estudio son Theta S (*dual-fisheye*), que proporciona una imagen equirectangular, Microsoft Azure (cámara RGB-D con amplio campo de visión), T265 (lente *fisheye* estéreo) y D435i (cámara RGB-D). De este estudio, Chappellet et al. concluyen, entre otras cosas, que la localización con la cámara de mayor campo de visión, Theta S, es la que menos se ve afectada por los cambios en el entorno. Dado que el método evaluado para estimar la pose se basa en características, las imágenes proporcionadas por esta cámara permiten extraer un número de características no alteradas más elevado.

Los sistemas de visión catadióptricos se componen de una cámara perspectiva y elementos reflectantes (espejos) y refractivos (lentes) [128]. El elemento reflectante de los sistemas catadióptricos puede estar formado por espejos planos, espejos con curvatura radial o espejos cónicos. Entre los segundos, se encuentran los espejos elípticos, hiperbólicos y parabólicos, que son de superficie cuadrática y además los tres más empleados [129].

Algunos de los sistemas de visión catadióptricos presentan la propiedad de que todos los rayos de proyección intersecan en un único punto común, denominado centro de proyección o punto de vista efectivo. A estos se les conoce como sistemas centrales. Sin embargo, este hecho no siempre se da como consecuencia de errores durante la fabricación del espejo o desalineación entre el eje del espejo y la cámara, lo cual puede ocurrir en los sistemas centrales. Además de ello, por su geometría, no todos los espejos son capaces de proporcionar un sistema de este estilo, por lo que también encontramos sistemas no centrales. En resumen, los sistemas de visión catadióptricos se pueden clasificar en centrales (un único centro de proyección) o no centrales (no presentan un punto de vista efectivo común).

Otro sistema de visión capaz de proporcionar una imagen panorámica es el sistema óptico *Panoramic Annular Lens* (PAL) [130], que cada vez se está empleando con mayor frecuencia [131-133]. Se compone principalmente de tres partes: en primer lugar, se encuentra el bloque de unidad principal PAL, el cual está basado en un sistema catadióptrico, donde se produce una primera refracción seguida de dos reflejos, lo que produce un ángulo más reducido que recibirá la segunda parte de esta sistema; en segundo lugar, este sistema dispone de un grupo de lentes de relé que permiten corregir las aberraciones y proporcionar una focal positiva; en tercer lugar, la última parte corresponde a un sensor. La imagen resultante presenta un área circular ciega en el centro debido a la oclusión producida por el segundo reflector. El campo de visión horizontal es de 360°, mientras que el campo de visión vertical viene dado por los ángulos (θ_{min}, θ_{max}), siendo el primero el mínimo ángulo vertical con el que se recibe un rayo incidente y el segundo el máximo ángulo vertical.

Hasta este momento todos los sistemas de visión descritos se componen de una única cámara, pero, como ya se ha comentado anteriormente, un sistema de visión omnidireccional puede estar formado por múltiples cámaras. En el mercado se pueden

encontrar varias cámaras polidióptricas como Facebook surround360 (con catorce lentes gran angular y tres lentes ojo de pez), Insta360 PRO (con seis lentes *fisheye*) y Ladybug6 (compuesta por seis lentes).

También encontramos cámaras que presentan una configuración más simple en cuanto a número de cámaras, son las denominadas cámaras de doble ojo de pez, como Samsung Gear 360, RICOH THETA Z1 o Garmin VIRB 360. Se encuentran compuestas por dos lentes *fisheye* que presentan un campo de visión cada una mayor a 180° y se encuentran posicionadas *back-to-back*. Con este sistema, se consigue cubrir la esfera completa, es decir, todo del entorno. Además, la diferencia con respecto a las cámaras mencionadas en el párrafo anterior es que únicamente se procesan dos imágenes, por contra, estas imágenes presentan una elevada distorsión por lo que deben ser preprocesadas. Algunas de las cámaras comerciales que se han mencionado tanto en este párrafo como en el anterior se muestran en la figura 2.4.

(a) (b) (c) (d)

Figura 2.4: Sistemas de imágenes esféricas comerciales: (a) Insta360 PRO diseñado por Insta360 [134], (b) Ladybug6 producido por FLIR [135] (c) RICOH THETA Z1 presentado por THETA [136] y (d) Garmin VIRB 360 diseñado por Garmin [137].

Como se verá en los procesos para obtener una vista panorámica, tanto en el caso de las imágenes obtenidas con un sistema catadióptrico como en el de las imágenes capturadas con una lente *fisheye*, ambas necesitan ser transformadas a un formato en el que la información obtenida sea más fácil de interpretar (*unwrapping*). Esto sucede porque, en ambos casos, la imagen que proporciona es circular. Por lo tanto, se realiza una transformación de coordenadas esféricas a cartesianas.

2.3.2 Obtención de una imagen panorámica

Una imagen panorámica se caracteriza por tener más información del entorno (es decir, un mayor campo de visión), que una imagen adquirida por una única cámara estándar. De hecho, suelen disponer de un campo de visión horizontal de hasta 360°. Las imágenes panorámicas son empleadas en numerosas aplicaciones, como realidad virtual, medicina, *street view*, o en automóviles, entre otras.

Como se ha detallado anteriormente, existen distintas configuraciones que permiten adquirir la información necesaria para construir una imagen de este tipo. Sin embargo, algunas de ellas requieren de un procesamiento, más o menos complejo, para conseguir una vista de 360° en una única imagen. En la tabla 2.2 se resumen algunas de las propiedades que se detallan a continuación.

Tabla 2.2: Comparativa de las diferentes formas de obtener una vista 360° en una única imagen.

Configuración sistema	Simplicidad procesamiento	Transformar a otro formato	Alineamiento + fusión	Resolución
Cámara catadióptrica	↑	Proyección cilíndrica	-	↓
Múltiples cámaras	↓	-	✓	↑
Cámara con doble lente *fisheye*	↓↓	Proyección esférica	✓	↑

El procesamiento más simple se produce cuando no se requiere unir imágenes tomadas desde distintos puntos de vista o dispositivos. Dentro de esta categoría se encuentran las cámaras catadióptricas, pues proporcionan una única imagen con este campo de visión. No obstante, por la forma en la que este sistema captura el entorno, el procesamiento consiste en expresar dicha imagen polar en una imagen panorámica a partir de un tipo de proyección, siendo la proyección cilíndrica la más empleada para esta configuración. En este caso, la altura del cilindro vendrá determinada por la región de interés de la imagen escogida, seleccionando el radio superior y el inferior. Dong et al. [138] analizan los métodos de *unwrapping* de una imagen capturada por un sistema de visión catadióptrico con punto de vista único, en el que se proyecta sobre una superficie cilíndrica y perspectiva. Chong et al. [139] proponen una algoritmo para un sistema de visión omnidireccional con punto de vista no único. La imagen panorámica final puede ser cilíndrica, cuboide o vista del plano del suelo.

Sin embargo, el problema que se encuentra en estas imágenes panorámicas es su baja resolución. La forma de aumentar el campo de visión sin perder resolución es combinando imágenes tomadas desde distintos puntos de vista y que presenten zonas comunes. Como consecuencia, el algoritmo necesario para adquirir la imagen panorámica requiere de más pasos [140], pues las imágenes no pueden unirse directamente. Cabe destacar que, además de que el procesamiento requiere de más pasos, también presenta un número de desafíos más elevado puesto que se pueden producir ciertos efectos no deseados como el de desenfoque o fantasma debido al paralaje y/o a una escena no estática (cuando las imágenes son capturadas en diferentes instantes de tiempo); así como una unión visible entre las imágenes causada por variación de exposiciones.

El algoritmo para unir dos o más imágenes capturadas desde distintos puntos de vista ya sea con la misma cámara o con otra, se compone principalmente de un proceso de alineamiento y otro de unión. En [141], se realiza una revisión de los algoritmos para estos dos procesos.

Los métodos de alineamiento de imágenes pueden dividirse en dos grupos: métodos basados en píxeles (o directos) y métodos basados en características. Mientras los segundos extraen características de las imágenes para estimar la transformación entre ellas, los primeros comparan las imágenes completas para hallar la transformación de

forma que se minimicen las diferencias píxel a píxel.

Los métodos directos pueden estar basados en la minimización de la diferencia de intensidad, en técnicas de correlación o en información común. Meneghetti et al. [142] utilizan métodos basado en la correlación para estimar el desplazamiento entre pares de imágenes. Después, calculan la matriz de homografía. Las imágenes que usan han sido capturadas rotando la cámara un ángulo muy pequeño sobre el eje vertical que pasa por el centro óptico. Estos métodos, dado que comparan píxel a píxel, suelen ser complejos, además de sensibles ante cambios de escala y rotaciones. [143]

En cuanto a los métodos basados en características, se fundamentan en la extracción de ciertas características distintivas de las imágenes (como puntos, líneas o esquinas) y en el establecimiento de correspondencias con otras imágenes a partir de sus descriptores para, finalmente, estimar un mapeo imagen a imagen para el registro [144]. Para ello, las características locales deben ser robustas ante cambios de escala, translación y rotación [145].

Sharma et al. [146] analizan el efecto de los detectores y descriptores locales en la imagen final tras aplicar un algoritmo de unión. En el estudio que exponen los autores, las combinaciones de detectores-descriptores se comparan en cuanto a número de pares de correspondencias, medidas de calidad (basadas en *Peak Signal to Noise Ratio* (PSNR), *Structural Similarity Index Measure* (SSIM), *Feature Similarity* (FSIM) y *Visual Saliency-Induced Index* (VSI)) sobre la imagen resultante y el tiempo de ejecución. Siendo la combinación de AKAZE como descriptor y AKAZE como detector la solución idónea en casi todas las situaciones estudiadas.

En la parte de registro de imagen, el objetivo es encontrar la transformación entre imágenes con el fin de alinearlas. Entre las transformaciones geométricas en el plano 2D se encuentran [42]: translación, transformación rígida (rotación y translación), semejanza, afín - que mantiene las líneas paralelas - y proyectiva (homografía) - conservando las líneas rectas. La transformación más empleada para alinear imágenes es la matriz de homografía. En esta línea, DeTone et al. [147] presentan HomographyNet, red que estima los ocho parámetros de la matriz de homografía relativa entre el par de imágenes de entrada. Se compone de una red de regresión y de una red de clasificación. Yan et al. [148] resumen diferentes técnicas basadas en el aprendizaje profundo que han sido propuestas para realizar varias de las etapas del algoritmo de unión de múltiples imágenes. Se encuentran métodos para resolver la coincidencia de imágenes, la estimación de homografía y la síntesis de imágenes.

En los ejemplos expuestos hasta el momento, se utilizan imágenes tomadas por una cámara estándar desde distintos puntos de vista o que realiza una rotación.

Analizando estos sistemas para la generación de una imagen panorámica, encontramos, en primer lugar, que en el caso del sistema compuesto por una única cámara que realiza una rotación, no hay presencia de distorsión y todas las imágenes son tomadas con el mismo dispositivo. Estas presentan ventajas en cuanto al procesamiento. Pese a ello, también presenta desventajas como que el entorno y el robot deben permanecer estáticos; el hecho de que hayan objetos en movimiento (personas, ramas de los árboles, nubes, etc), hace que sea mucho más complejo e incluso que aparezcan ciertos efectos que disminuyen la calidad de la imagen resultante. Alomran y Chai [149]

emplean un algoritmo basado en características para unir imágenes que han sido adquiridas rotando una cámara situada sobre un trípode estático. Para el registro, obtienen una transformación rígida que únicamente se compone de la matriz de rotación.

En cuanto a la generación de imágenes panorámicas a partir de dos imágenes *fisheye* capturadas con dirección opuesta, se requiere de una transformación a una proyección en la que sea más fácil unir el par de imágenes, pues con su distorsión original esta tarea no es posible.

Con respecto a obtener la transformación entre el par de imágenes, podemos encontrar varias propuestas. Ho y Budagavi [150] sugieren calcular una matriz afín a partir de puntos coincidentes seleccionados de forma manual. Tras aplicar esta transformación, proponen aplicar otra segunda matriz afín que se compone de un desplazamiento, el cual ha sido obtenido tras una búsqueda en la región de solape, de coincidencia de plantilla con objetos. Más tarde, con el objetivo de mejorar el algoritmo anterior, Ho et al. [151] proponen conseguir la alineación entre ambas imágenes mediante la interpolación de cuadrícula basada en mínimos cuadrados móviles rígidos (MLS). Souza et al. [152] sugieren realizar, en primer lugar, una coincidencia de plantillas, las cuales son agrupaciones de puntos ORB. A partir de los desplazamientos obtenidos durante las correspondencias de todas las plantillas, se estima la matriz de homografía. Los trabajos descritos intentan alinear ambas imágenes en 2D, sin embargo, Ni et al. [153] lo hacen en 3D, estimando una matriz de rotación entre las coordenadas en la esfera unitaria.

2.4 Métodos de extracción de información visual

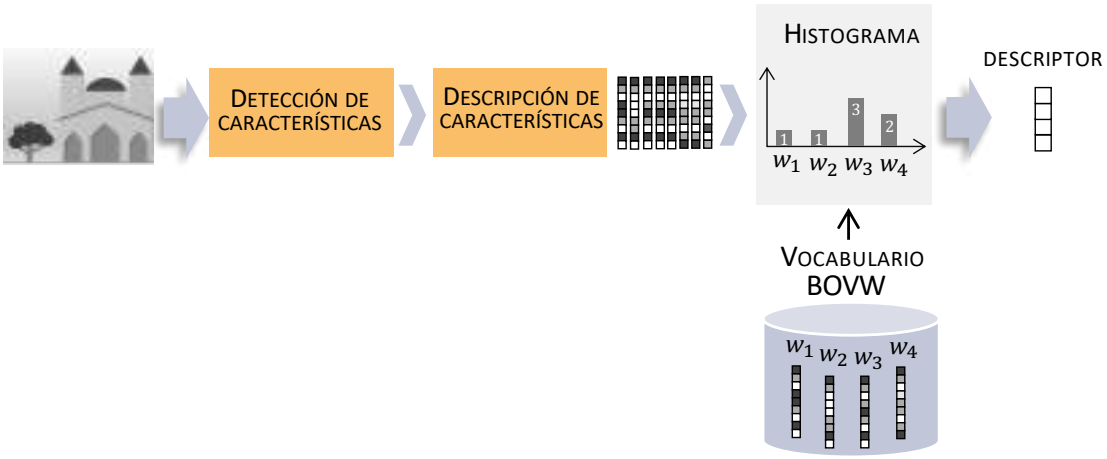
Una imagen contiene una gran cantidad de datos, triplicándose su dimensión en el caso de ser a color, lo que conlleva un alto coste computacional para todos los procesamientos necesarios. Además, en ocasiones se requiere de imágenes con alta resolución, pues proporcionarán mayor detalle del entorno, pero como consecuencia se necesitará un tiempo aun mayor para su procesamiento.

Por lo que respecta a los métodos de extracción de características, se emplean para reducir la dimensión de una imagen seleccionando únicamente información distintiva. De esta forma, una imagen con una gran cantidad de píxeles es representada por un único descriptor (características globales, ver figura 2.5(a)) o por un conjunto de descriptores (características locales, ver figura 2.5(b))). Asimismo, también se puede utilizar la técnica de bolsa de palabras para representar la información visual (ver figura 2.5(c)). En la misma, cada imagen es representada por un único vector que corresponde a un histograma de frecuencia que se forma como resultado de agrupar características locales. Se suele emplear en reconocimiento de imágenes [154, 155] y detección de cierre de bucle en SLAM [156, 157]. Por ejemplo, Chen et al. [158] proponen, para mejorar el rendimiento del *backend* de un sistema de SLAM con LiDAR, emplear también información visual. Concretamente, la detección del cierre del bucle se resuelve a partir de la coincidencia entre características visuales (bolsa de palabras) y las características geométricas (nubes de puntos proporcionada por LiDAR). En algunos trabajos, como veremos a continuación, clasifican esta técnica como descriptor de apariencia global, pues cada imagen tiene asignado un único vector descriptor. Por el contrario, en otros,

la consideran como apariencia local, ya que se basa en descriptores de características locales.

(a) Características globales

(b) Características locales



(c) Bolsa de palabras visuales

Figura 2.5: En (a) se puede ver como una imagen es representada por un único vector descriptor, por el contrario, en (b), para una misma imagen, se tiene un conjunto de descriptores. En (c), la imagen es descrita por un único descriptor que se obtiene a partir del histograma de frecuencia de palabras visuales, que son descriptores de apariencia local.

Por un lado, los métodos basados en apariencia global muestran cierta sensibilidad al ruido o a la variación en la iluminación o la escala. A pesar de esto, en comparación con las características locales, requieren de un menor tiempo de cálculo (lo que resulta importante para tareas que se llevan a cabo en tiempo real), y una menor memoria (pues cada imagen tiene asociado un único descriptor). Además, también funcionan mejor en imágenes con baja textura en comparación con las de características locales.

Por otro lado, a diferencia de los métodos globales, las técnicas basadas en características locales no requieren ningún paso de segmentación previo y tienden a presentar mayor robustez ante oclusiones [159].

Tras todo lo visto, hemos de señalar que los enfoques globales presentan diversas limitaciones o desventajas que, desde la perspectiva de los enfoques locales, aparecen como ventajas, y viceversa. Por dicho motivo, se suelen combinar para realizar ciertas tareas [160-163].

Por ejemplo, Su et al. [164] proponen usar ambos tipos de características para el reconocimiento de caras basándose en el hecho de que los humanos identificamos las caras fijándonos tanto en los rasgos globales de la cara (pelo, contorno, ropa, etc) como en detalles locales (ojos, nariz, boca). Con respecto a la tarea de localización, Fang et al. [133] plantean el uso de un algoritmo que combina ambos tipos de características y se compone de dos etapas. En la primera etapa, denominada localización gruesa, se emplean descriptores NetVLAD, red troncal (CNN, como AlexNet o VGG16) + capa NetVLAD, para obtener las k imágenes más similares en un conjunto de imágenes del entorno disponible. Después, en la segunda etapa (denominada localización fina), se obtiene la imagen más similar a la de consulta entre las obtenidas en la etapa anterior. Todo ello mediante descriptores de puntos clave (ORB, SIFT y GeoDesc) y la matriz fundamental junto con RANSAC. La imagen que presente más *inliers* con la de consulta se determinará como la más similar.

Como se tendrá ocasión de comprobar a lo largo de esta sección, los métodos de extracción de características, ya sean locales o globales, se dividen en técnicas tradicionales y técnicas que utilizan redes neuronales. La principal ventaja de estas últimas con respecto a las primeras es que se pueden entrenar bajo diferentes condiciones, de tal forma que suelen ser más robustas en situaciones que generan desafíos. Sin embargo, esto también conlleva la necesidad de llevar a cabo un proceso de entrenamiento con un *dataset* que debe estar disponible de antemano, lo cual puede resultar laborioso.

2.4.1 Características globales

El enfoque basado en características globales es más compacto debido a que la cantidad de información que se encuentra en una imagen viene representada por un solo vector, necesitando, en consecuencia, menos tiempo computacional y menos memoria para almacenar información del entorno. En este caso, se describe todo el contenido de la imagen (en cuanto a textura, forma o color), por lo que se pierde información local del entorno, pero se mantiene la relación entre toda la información capturada en la imagen, ya que describen la imagen en global teniendo en cuenta todos los píxeles. En ocasiones, se hace referencia al descriptor global como descriptor holístico.

Uno de los métodos más empleados es el histograma de gradientes orientados (HOG, *Histogram of Oriented Gradients*) [165] que se compone concatenando todos los histogramas de las n diferentes celdas en las que se divide la imagen. Por lo tanto, cada celda es representada por un histograma y este cuantifica, en b *bins*, la orientación del gradiente calculado para cada píxel dentro de dicha celda. Finalmente, el resultado es un descriptor con una dimensión y una longitud igual a $n \cdot b$.

Otro descriptor que también se usa con frecuencia es Gist [166], que se basa en filtros de Gabor, con m orientaciones, que se aplican sobre la imagen en varios niveles de resolución. Después, cada una de estas imágenes filtradas se divide en varias celdas y se calcula el valor medio de los píxeles dentro de cada celda. El descriptor final se forma

concatenando los valores medios obtenidos en las diferentes escalas y orientaciones.

En cuanto al empleo de *deep learning*, las redes neuronales convolucionales son capaces de extraer, de imágenes sin procesar, características abstractas de alto nivel. Por dicho motivo, se comenzó a usar para obtener representaciones de imágenes. En cuanto al modelo en el que se basan, suelen ser redes neuronales convolucionales estándar (ya entrenadas) o personalizadas. En lo referente a la generación de un descriptor de apariencia global, hay dos posibles alternativas: utilizar la salida de una de las (I) capas *fully connected* (por ejemplo en [167]) o (II) bien utilizar la salida de capas convolucionales (como en [168, 169])

Por un lado, la salida de una capa *fully connected* puede entenderse como un descriptor que representa la imagen de entrada. Cabrera et al. [170] realizan *image retrieval* comparando descriptores de apariencia global que corresponden a la salida de una de las capas *fully connected* de la red AlexNet.

Por otro lado, el descriptor puede adquirirse a partir de un mapa de características obtenido a la salida de una capa convolucional. Para ello este se interpreta como el resultado de dividir la imagen en diferentes celdas del mismo tamaño y calcular sus correspondientes descriptores, el conjunto de los cuales forma el mapa de características. Tras concatenar todos estos descriptores, se obtiene el descriptor único. Este último procedimiento se asemeja mucho a los métodos tradicionales descritos anteriormente. Arroyo et al. [171] sugieren que, en vez de generar el descriptor a partir de una capa convolucional, este sea generado fusionando la información proporcionada por varias de estas capas, consiguiendo así que sea más eficiente. Tras concatenar las características vectorizadas obtenidas por cada una de estas capas convolucionales, el vector obtenido presenta una longitud bastante extensa. Por dicho motivo, los autores llevan a cabo una compresión de los datos redundantes que proporcionan las capas convolucionales, consiguiendo un descriptor final con menor longitud y sin disminuir la precisión. Además de esto, también realizan una binarización.

Con respecto a la capa adecuada para generar el descriptor, Sünderhauf et al. [172] estudiaron la robustez de representaciones obtenidas a partir de diferentes capas de la arquitectura de AlexNet. Principalmente el estudio se centra en el cambio de punto de vista y de apariencia del entorno. Los autores concluyen que las salidas de las capas intermedias proporcionan mayor robustez ante cambios de apariencia. Por el contrario, las capas más cercanas a la salida de la red hacen que sean más robustas a los cambios de punto de vista.

Li et al. [173] evalúan quince métodos de extracción de características globales para el reconocimiento de lugares. Los métodos evaluados se dividen en tres clásicos (como HOG y Gist), seis basados en redes neuronales conocidas (como AlexNet [174] o ResNet50) y seis basados en redes neuronales que los autores han adaptado. Cabe destacar que dos de los métodos que evalúan y comparan son dos variaciones del modelo de bolsa de palabras visuales: empleando ORB (se encuentra dentro de los métodos tradicionales) y SuperPoint (esta combinación es uno de los últimos seis métodos). Los autores determinan que ambos modelos de bolsa de palabras únicamente funcionan bien cuando la variación en apariencia global es pequeña. Los descriptores globales basados en redes neuronales recuperan más coincidencias en entornos más complejos

y cambiantes, a costa de un coste computacional más alto.

2.4.2 Características locales

Las características locales engloban una primera parte de detección y una segunda de descripción. En cuanto a los descriptores, estos se pueden clasificar, según el formato de representación de los valores, en coma flotante o binario.

Los detectores se pueden definir por el tipo de característica (como esquina o *blob*) o por el tipo de invariabilidad [175]. Por un lado, una esquina se puede definir como el punto de intersección de dos líneas de borde o el punto en el plano imagen en el que existen dos orientaciones de gradiente dominantes y diferentes [176]. En cuanto a los métodos de detección de esquinas que existen se pueden agrupar, tal y como se detalla en [177], en función de si se basan en la variación local de la intensidad, en la información de la forma del contorno de los bordes o en un modelo. Por otro lado, un *blob* es una región en la que todos los píxeles son similares entre sí pero difieren de los vecinos circundantes [176]. En esta línea, los detectores de *blobs* localizan estructuras que son máximas en el espacio de escala de la imagen, el cual ha sido generado con filtros lineales o no lineales. De este modo, este tipo de característica viene determinado por sus coordenadas (x, y) así como por la escala σ. Otra forma de detectar este tipo de características en mediante segmentación (como MSER [178]) en cuyo caso lo que se detecta es una región de interés. Mientras que con los detectores de *blobs* se tiende a obtener un mayor número de características locales que con los detectores de esquinas, es cierto que estos últimos proporcionan características locales más estables y repetibles [159].

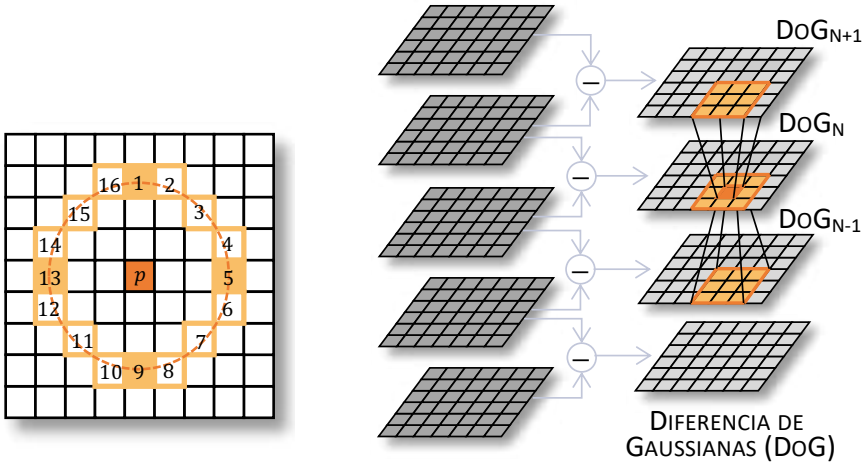
Además de estos dos tipos de características, también se pueden detectar bordes (por ejemplo, con el operador de Canny [179]). Li et al. [176] han realizado un estudio que emplea diferentes métodos de extracción, mientras que Huang et al. [180] realizan un estudio de los métodos de descripción de características.

2.4.2.1 Métodos tradicionales

Se busca que tanto los detectores como los descriptores presenten ciertas propiedades tales como: robustez, repetibilidad, precisión, eficiencia, cantidad, carácter distintivo/informativo, invarianza y localidad. Todas estas propiedades se encuentran detallas en [181] y [182]. Para ofrecer una solución adecuada al problema que se pretende resolver con características visuales, resulta fundamental escoger un método que sea invariante a las condiciones concretas de la aplicación. A continuación, se van a describir algunos de los métodos propuestos.

Entre los métodos que solo detectan (y no describen) puntos característicos locales, se encuentra FAST (acrónimo del inglés *Features from Accelerated Segment Test*) que fue propuesto por Rosten y Drummond en [5]. Este método se caracteriza por comparar la intensidad de los píxeles situados sobre una circunferencia alrededor del píxel candidato a esquina (p), como se muestra en la figura 2.6(a). Este píxel ubicado en el centro se clasifica como esquina si n de los píxeles localizados en el circulo son más brillantes (con intensidad mayor que la intensidad del píxel central más un umbral) o más oscuros (con intensidad menor que la intensidad del píxel central menos

un umbral) o, en caso contrario, se descartará. Para que esta comprobación sea más rápida, primero se compara con cuatros píxeles que corresponden a los índices 1, 5, 9 y 13. Si al menos tres de estos píxeles cumplen con lo mencionado, entonces se realiza la comprobación con los píxeles restantes.



(a) (b)

Figura 2.6: Procedimiento para determinar si un punto (■ es característico con: (a) el método FAST y (b) el método SIFT).

Algunos autores proponen técnicas basadas en características locales que detectan y extraen descriptores: SIFT (acrónimo del inglés *Scale Invariant Feature Transform keypoint*) [183], SURF (acrónimo del inglés *Speeded Up Robust Features*)[14], KAZE [41] y ORB (*Oriented FAST and Rotated BRIEF*) [11].

SIFT. El primero de los mencionados, fue propuesto por Lowe [183] y se basa en un espacio de escala compuesto por diferencia de gausianas. En primer lugar, se genera una pirámide de resolución múltiple sobre la imagen de entrada y posteriormente se calcula la diferencia en píxeles de dos escalas de niveles consecutivos, D(x,y,σ). En segundo lugar, se buscan extremos locales en una ventana de 3x3x3 píxeles en el espacio de escala. Por lo tanto, será una característica local si es un máximo o mínimo local en comparación con sus 8 vecinos más cercanos de ese nivel, así como con sus nueve vecinos más cercanos superiores e inferiores. Todo esto se muestra en la figura 2.6(b). Finalmente, el resultado de la parte de detección, será el punto característico $p(x, y, \sigma_i)$ que vendrá definido por su posición y por la escala en la que se detectó σ_i.

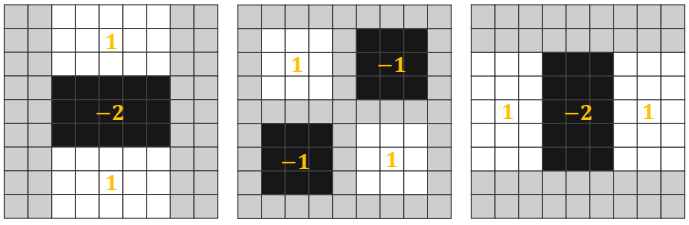
Para el descriptor, se escoge una región de 16x16 píxeles alrededor de la posición del punto característico detectado. Después, se normaliza en función de la escala y la orientación dominante. Con estos objetivos, en primer lugar, se calcula el gradiente y la región se divide en subregiones de 4x4 píxeles. Para cada una de estas subregiones, se crea un histograma en el que se cuantifican las orientaciones en 8 posibles grupos (360° separados por 45°). Finalmente se obtiene, como resultado de concatenar los 16 histogramas de 8 *bins*, un descriptor de 128 (16x8) elementos.

SURF. Como una alternativa más rápida, Bay et al. [14] propusieron este método. En este caso, la detección se realiza aplicando el determinante del operador Hessiana

en cada imagen que compone el espacio de escala. Para una posición $\mathbf{x} = (x, y)$ en la escala σ, la matriz Hessiana se calcula como:

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2}{\partial x^2} g(\sigma) & \frac{\partial^2}{\partial x \partial y} g(\sigma) \\ \frac{\partial^2}{\partial x \partial y} g(\sigma) & \frac{\partial^2}{\partial y^2} g(\sigma) \end{bmatrix} \tag{2.1}$$

donde cada elemento corresponde a derivada parciales gaussianas de segundo orden. Los máximos del determinante de la matriz Hessiana determinarán la ubicación (x, y) y la escala σ. En cuanto a la generación del espacio de escala, se genera aplicando filtros de *box*, que son derivadas Gaussianas aproximadas de segundo orden (ver figura 2.7), con diferentes tamaños sobre la imagen original. De esta forma, se evita el *aliasing* al no tener que reducir la resolución de la imagen.



(a) (b) (c)

Figura 2.7: Filtros de *box*. Aproximaciones de las derivadas parciales Gaussianas de segundo orden: (a) yy, (b) xy y (c) xx.

Para estimar la orientación, se considera la región circular con centro en el punto de interés y con radio igual a $6s$. Esta región se recorre en ventanas con un ángulo de 60° y en cada una de ellas se calcula la suma de todas las respuestas de *Haar-wavelet* (en dirección x e y). El vector de orientación local con mayor longitud determinará la orientación.

Para la obtención del descriptor, en primer lugar, se selecciona una región cuadrada de tamaño $20s$ x $20s$ alrededor del punto de interés y con orientación igual a la estimada. Esta región se divide a su vez en 4 x 4 cuadrados. En cada uno de estos cuadrados se obtiene el sumatorio de cada una de las respuestas *wavelet* de *Haar* ($\sum d_x$ y $\sum d_y$) calculadas para cada zona de 5 x 5. Además, también se considera la suma de los valores absolutos de las mismas ($\sum |d_x|$ y $\sum |d_y|$). Tras concatenar estos cuatro valores para todos los cuadrados, se obtiene finalmente un vector descriptor de longitud igual a $(4 \cdot 4) \cdot 4 = 64$.

KAZE. Alcantarilla et al. [41] presentaron una mejora de los anteriores métodos llamada KAZE. En este caso, el algoritmo funciona en un espacio de escala no lineal, a diferencia de los dos que hemos visto anteriormente. Con dicha propuesta, se pretende mejorar el carácter distintivo y la precisión durante la detección, ya que los anteriores se ven afectados cuando se crea un espacio de escala lineal, debido tanto al ruido como a que los bordes son suavizados de forma indistinta. Así, el desenfoque se adapta localmente a los datos de la imagen y, en consecuencia, se suaviza el ruido, pero no los bordes.

ORB. Rublee et al. [11] proponen un algoritmo que combina un detector FAST [5] mejorado y el descriptor BRIEF (acrónimo del inglés *Binary Robust Independent*

Elementary Features) [2] pero solucionando la falta de invariabilidad ante rotación de este. En primer lugar, como se ha comentado, la detección se basa en el método de extracción de características FAST, escogiendo un radio de 9 píxeles (FAST-9). Sin embargo, FAST presenta algunas limitaciones: no produce características en múltiples escalas; tampoco funciona correctamente en esquinas generadas por bordes perfectamente alineados con los ejes x, y ni tiene un componente de orientación. Ante esto, el algoritmo de ORB crea una pirámide de escalas de la imagen en la que se detectan características FAST. Para seleccionar los N mejores puntos clave, se ordenan en función de la medida del detector de esquinas de Harris. En lo que respecta a la orientación, su obtención se basa en el centroide de intensidad, de modo que, la orientación θ se calcula como el $atan2$ del vector que va desde el centro de la esquina hasta el centroide que se puede obtener a partir de los momentos de la región considerada.

2.4.2.2 Métodos basados en redes neuronales

En cuanto a las técnicas basadas en redes neuronales, las podemos clasificar en función de si solo extraen características locales (detector), como *Quad-networks* [184] (no supervisado) o TILDE (acrónimo del inglés *a Temporally Invariant Learned Detector*) [15] (supervisado), si solo calcula vectores característicos (descriptores), como [185], [186], [187], L2-Net [188] GeoDesc [189] o si realiza ambos procesos (detector y descriptor). Algunos de estos últimos se describirán brevemente en esta sección, pero antes cabe comentar que aunque FAST use *Machine Learning*, no se ha incluido en este apartado porque sólo lo hace para acelerar el proceso de detección de las esquinas.

Liu et al. [190] han realizado varios experimentos con el objetivo analizar y comparar distintos métodos bajo diferentes condiciones en la imagen (compresión, iluminación, desenfoque y cambios de punto de vista). Por un lado, evalúan y comparan métodos de detección tradicionales (Harris [191], FAST [5] - BRISK [192], ORB [11], SIFT [183] - SURF [14] y KAZE [41]) y basados en redes neuronales (DetNet, LIFT [8], multiscale[193], TILDE [15], y SuperPoint [147]). Por otro lado, evalúan y comparan métodos de descripción tradicionales (BRISK, FREAK, BRIEF, SURF, ORB, KAZE y SIFT) y basados en redes neuronales (Pre-Net, Siamese-Net, Triplet-Net, LIFT y SuperPoint).

Ma et al. [194] realizaron una revisión bastante exhaustiva de los métodos existentes de emparejamiento de imágenes, pero también de los métodos de detección y descripción de características. Todo ello pasando por los métodos más tradicionales hasta los más recientes que emplean redes neuronales.

A continuación, se describen algunos métodos basados en redes neuronales que proporcionan el punto característico así como el descriptor. Además, en la tabla 2.3 se encuentran las principales características de estos métodos y los basados en redes neuronales que se describen en el apartado 2.4.2.3.1.

LIFT (*Learned Invariant Feature Transform*). Yi et al. [8] proponen este método que consta de un proceso compuesto principalmente por tres elementos interconectados donde cada uno se basa en redes neuronales convolucionales (CNN). En el artículo, identifican cada uno de ellos como *Detector*, *Orientation Estimator* y *Descriptor*. En primer lugar, se detectan las posiciones de los puntos característicos mediante la

primera red convolucional y *Non-local Maximum Suppression* (NMS). Después, se obtienen regiones centradas en los máximos locales obtenidos para que tanto *Orientation Estimator* como *Descriptor* se centren en ellos, ya que, como veremos a continuación, el entrenamiento se basa en regiones. Por lo tanto, antes de pasar a la siguiente etapa emplean una capa de Transformación Espacial.

En la parte de entrenamiento se emplea una arquitectura siamesa compuesta por cuatro ramas. Para el entrenamiento utilizan puntos característicos SIFT obtenidos en el marco de una aplicación *Structure from Motion*.

Superpoint. DeTone et al. [195] proponen un algoritmo que utiliza toda la imagen en vez de parches como otros enfoques. En cuanto a la arquitectura, en primer lugar, para reducir la dimensionalidad de la imagen de entrada, se encuentra un *encoder* (VGG-style) el cual se compone de capas convolucionales, reducción de muestreo espacial mediante agrupación y funciones de activación no lineal. La detección de puntos característicos y la descripción se lleva a cabo de forma paralela, es decir, en un único paso, por lo que el tensor obtenido será la entrada a dos ramas en las que se llevará a cabo cada una de estas tareas y que se componen por *decoder*.

Se trata de un enfoque autosupervisado, por lo que genera un *pseudo-ground truth* de puntos característicos. Para ello, entrenan previamente un detector base (MagicPoint) usando un conjunto de imágenes sintéticas que contienen diferentes formas geométricas (cubos, triángulos, tablero de ajedrez, etc) con sus respectivas etiquetas de las posiciones de las esquinas. Con el fin de mejorar el rendimiento del detector, así como la repetibilidad de la detección de puntos de interés, los autores de este trabajo [195] proponen utilizar el detector base de forma conjunta con una adaptación homográfica (multiescala y multihomografía) para generar las etiquetas de los puntos característicos en imágenes en las que no se dispone de esta información. En la adaptación homográfica, dada una imagen se aplican diferentes homografías de forma aleatoria y se detectan los puntos característicos mediante MagicPoint. Tras aplicar la transformación inversa a dichos puntos, se obtienen en la imagen de entrada. La función de pérdida se define como la suma de la función de pérdida correspondiente al detector y la de la parte del descriptor para la cual se utiliza una arquitectura siamesa.

Unsuperpoint. Christiansen et al. [196] presentan esta arquitectura que toma una imagen de color como entrada y genera un mapa de características que será procesado en tres submódulos. Cada submódulo está formado por dos capas convolucionales con una dimensión de 256 y diferentes canales para cada uno de los submódulos: un canal para el submódulo de *score*, 2 canales para el de las posiciones y 256 para el de descriptores. Tanto el submódulo de puntuación (*score*) como de posición tienen, después de la última capa, una función de activación sigmoidea con el fin de obtener ambas predicciones entre 0 y 1. Dado que la altura y la anchura de la imagen de entrada se reduce un factor de ocho, cada región de 8x8 en la imagen de entrada será una entidad a la salida. Las salidas de los tres submódulos se ordenan por puntuación más alta.

Además de todo lo anterior, Christiansen et al. [196] proponen usar regresión para las posiciones de los puntos, incorporando de este modo una supresión de no máximos, y una nueva función de pérdida para que las predicciones se distribuyan uniformemente. Con todo esto, a diferencia de las arquitecturas anteriores, el entrenamiento se realiza

en una única ronda y no necesita generar un conjunto de *pseudo-ground truth* de puntos ni tampoco puntos característicos generados a partir de *Structure-from-Motion* (SfM). Como consecuencia, se emplea un esquema de entrenamiento autosupervisado compuesto por una red siamesa que aprende de forma simultánea las tres tareas. En una de las ramas se procesa la imagen de entrada sin transformación espacial mientras que en la otra rama la imagen de entrada es transformada por una homografía aleatoria. Antes de usar la arquitectura de Unsuperpoint en cada rama, a ambas imágenes se les aplica transformaciones no espaciales como brillo y ruido.

D2-NET. Dusmanu et al. [197] sugieren utilizar una única red neuronal convolucional ($\mathcal{F}$) para obtener un conjunto de mapas de características (un tensor 3D) que se empleará tanto para la descripción como para la detección de puntos característicos. Dado un tensor 3D, $\mathcal{F}(I) \in \mathbb{R}^{hxwxn}$, se puede interpretar como un conjunto de hxw vectores descriptores con longitud n o un conjunto de n mapas 2D de respuesta del detector. La primera interpretación se emplea para la descripción y la segunda para la detección. El tensor 3D corresponde a la salida de la capa conv4_3 de una red neuronal VGG16 la cual se ha entrenado con el *dataset* de ImageNet.

En cuanto a los datos de entrenamiento, se seleccionan pares de imágenes que presenten un porcentaje alto de solape basándose en una nube de puntos SfM dispersa. De todos los puntos 3D comunes, se selecciona uno de forma aleatoria y sus proyecciones en ambas imágenes serán el centro de una región de 256x256 píxeles que será la entrada para el entrenamiento. Debido a la arquitectura de la red, el tensor 3D tendrá una resolución igual a 32x32 (una octava parte de la región de entrenamiento), por lo que cada posición 2D en este tensor corresponderá a un área de 8x8 de la región inicial. La función de pérdida propuesta realiza un promedio ponderado de la *triplet margin* según su puntuación de detección, pero esta función de pérdida solo se aplica a las correspondencias entre ambas regiones. Para poder obtener este conjunto de correspondencias, se realiza una proyección desde el tensor de una de las regiones al otro tensor y aquellos puntos 2D que tengan un valor de profundidad similar (su diferencia sea menor a un determinado umbral) serán puntos coincidentes.

R2D2. Revaud et al. [198] presentan una red totalmente convolucional que obtiene, para una imagen de entrada, tres salidas: un tensor 3D, un mapa de calor y un mapa de fiabilidad. Atendiendo a su arquitectura, tiene una red troncal basada en la arquitectura de L2-Net [188], obteniéndose a la salida un tensor 3D con dimensión de 128, el cual será la entrada a dos ramas. En la primera, se realiza una L2-normalización, generando así el conjunto de descriptores (uno para cada píxel). En la segunda rama, cada elemento del tensor es elevado al cuadrado. Después de esta operación, se vuelve a dividir en dos ramas, ambas compuestas por una capa convolucional de 1x1 y una capa softmax. En la primera rama se obtiene el mapa de fiabilidad donde cada posición proporciona un valor estimado sobre la propiedad discriminatoria del descriptor en dicha posición. En la segunda, la salida es un mapa de calor que se entrena para que aparezcan máximos locales que son fuertes y repetibles.

El entrenamiento de R2D2 es autosupervisado, de tal forma que no se tiene conocimiento acerca de qué regiones son de interés. Para obtener los datos de entrenamiento, los autores del trabajo [198] proponen un *pipeline* en el que primero obtienen un conjunto de puntos 3D, así como la pose de la cámara. Además, observaron que, en lugar de

obtener la matriz fundamental directamente a partir de la pose de la cámara, presenta mayor fiabilidad estimarla a partir de las correspondencias 2D generadas mediante SfM. Finalmente, extraen las correspondencias densas mediante flujo óptico. Además de las correspondencias obtenidas con este proceso, también entrenan la red con correspondencias a partir de pares de imágenes con transformaciones conocidas. En cuanto al entrenamiento del descriptor, no utilizan una función de pérdida de tripleta estándar, sino que proponen emplear una aproximación de la métrica *Average Precision* (AP).

2.4.2.3 Métodos basados en imágenes omnidireccionales

Como ya se ha comentado, todos los métodos propuestos han sido diseñados para que sean invariantes ante ciertos cambios, como escala, iluminación o perspectiva, entre otros. Sin embargo, son pocos los que han tenido en cuenta la distorsión que puede aparecer en una imagen capturada por un sistema de visión con una lente *fisheye*. Este tipo de imagen se caracteriza por tener una distorsión diferente en función de la región dentro de una misma imagen, siendo menor en el centro y mayor conforme se encuentra más cerca de la periferia.

sRD-SIFT. Lourenco et al. [209] sugieren realizar ciertas modificaciones al algoritmo de SIFT [183] con el objetivo de tener una mayor repetibilidad y efectividad en presencia de distorsión radial. Lourenco et al. proponen emplear un filtrado adaptativo para generar la representación del espacio de escala y que compensa la distorsión local. El filtrado adaptativo se basa en una función kernel gaussiano con una desviación estándar que varía con el radio de la imagen del píxel.

Spherical ORB (SPHORB). En este caso, Zhao et al. [12] proponen SPHORB, el cual se basa en ORB y en una estructura de malla geodésica. En la parte de detección, se compara la intensidad de un determinado píxel con respecto a sus 18 píxeles más cercanos que se encuentran sobre un hexágono. Dicho píxel será detectado como esquina si al menos 10 de estos píxeles son más brillantes o más oscuros que el píxel central. Una vez detectada una esquina, se construye su correspondiente descriptor comparando la intensidad con los vecinos más cercanos dentro de un parche.

Fisheye Spherical Distorted BRIEF (FSD-BRIEF). Zhang et al. [210] sugieren utilizar un descriptor BRIEF y adaptarlo para imágenes *fisheye*. El proceso de obtención de este descriptor BRIEF distorsionado se compone de cuatro pasos. En el primero, se diseña la función de densidad de píxeles, la cual se define como el área de la región de un píxel sobre la esfera unidad, que se aproxima a un paralelogramo ya que su tamaño es lo suficientemente pequeño. De este modo, la función de densidad de un píxel depende de la función de mapeo de proyección esférica debido a que es calculada en la superficie de la esfera unitaria. Después, el segundo paso tiene como objetivo calcular el centroide gris 3D, el cual determina la dirección del descriptor como el área circular en la superficie esférica unitaria, cuyo centro es un punto característico obtenido con FAST y que ha sido reproyectado mediante el modelo de cámara fisheye. El tercer paso propuesto consiste en obtener una matriz de inclinación de puntos característicos, la cual consiste en una transformación de coordenadas y cuya finalidad es representar la posición y la dirección de un punto característico. En este paso, hay dos sistemas de coordenadas con el mismo punto de origen, el de la cámara y el de altitud del punto característico cuyo eje Z coincide con el vector que va desde el centro del sistema

Tabla 2.3: Comparativa métodos de detección y extracción de características locales basados en redes neuronales.

Método	LIFT [8]	Superpoint [195]	Unsuperpoint [196]	D2-net [197]	R2D2 [197]	FisheyeSuperpoint [199]	Tian et al. [200]
Tipo de imagen	pinhole	pinhole	pinhole	pinhole	pinhole	fisheye	fisheye
Dataset entrenamiento	"*Piccadilly Circus*" y "*Roman-Forum*" de [201]	MS-COCO [202]	MS-COCO [202]	MegaDepth [203]	"Aachen Day-Night dataset" [204] [205] y "Oxford and Paris retrieval dataset" [206]	Oxford RobotCar Dataset [207]	WoodScape [208]
Datos entrenamiento	Puntos característicos de SfM	MagicPoint + Homografías	Correspondencias sintéticas con homografía	Correspondencias reales basadas en información de profundidad	Correspondencias sintéticas (homografías) y reales (*optical flow*)	MagicPoint + *Fisheye warping* y *unwarping* (transformadas euclídeas)	Transformar la imagen *fisheye* original eliminando la distorsión, aplicar transformación homogénea y añadir la distorsión.
Entrada entrenamiento	Cuatro parches	Imagen completa	Imagen completa	Par de parches de $256x256$	Dos imágenes completas	Imagen completa	Dos imágenes completas
Arquitectura	Tres componentes en serie.	Un *encoder* seguido de dos *heads* de salida (detector y descriptor).	Un *backbone* seguido de tres *heads* de salida (puntuación, posiciones relativas y descriptores).	Un *backbone*	Un *backbone* seguido de una capa de normalización L2 (descriptores) y una operación de elemento cuadrático.	Un *encoder* seguido de dos *heads* de salida (detector y descriptor).	Un *backbone* seguido de tres *heads* de salida (puntuación, posiciones relativas y descriptores). Utiliza convolucionales deformables.

de coordenadas de la cámara hasta la proyección del punto característico en la esfera unidad. Para finalizar, en el cuarto paso, se extrae el descriptor. Para ello la plantilla de BRIEF es distorsionada empleando la matriz de inclinación calculada en el paso anterior, de este modo se ajusta a la forma de distorsión del área adyacente del punto característico.

2.4.2.3.1. Métodos basados en redes neuronales

.

En cuanto a las redes neuronales, cabe señalar que algunos de los métodos de extracción y descripción basados en esta técnica que se han expuesto anteriormente en este apartado hacen uso de la matriz de homografía para generar pares de imágenes transformadas. Así, simulan que han sido capturadas desde diferentes puntos de vista durante el entrenamiento con el fin de que sea invariante ante dicho aspecto. El problema es que la matriz de homografía no funciona correctamente en las imágenes de lente *fisheye* debido a su distorsión. Por ello, los siguientes trabajos proponen alternativas.

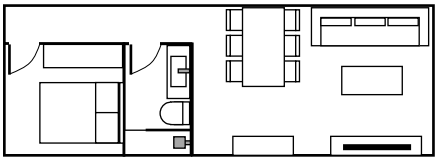
Konrad et al. [199] proponen FisheyeSuperpoint que consiste en una adaptación de SuperPoint [195] para este tipo de imágenes. Para ello, proponen reemplazar las transformaciones basadas en homografías por transformaciones aptas para las imágenes *fisheye*, es decir, transformaciones euclídeas. El procedimiento para ello consiste en, dada una imagen de lente *fisheye*, cada píxel es proyectado sobre una esfera unidad mediante una función de proyección no lineal. Después, para generar una pose virtual, se produce de forma aleatoria una matriz de rotación y un vector de traslación. Finalmente, se reproyectan sobre el plano imagen utilizando la función inversa de proyección empleada en el primer paso.

Tian et al. [200] proponen una arquitectura de red compuesta por una red troncal, obteniéndose a la salida un mapa de características con un tamaño igual a $1/8$ veces el de la imagen de entrada. Esta salida es procesada para obtener, finalmente, tres tensores del mismo tamaño que corresponden a la puntuación, posiciones relativas y descriptores. Por lo tanto, la posición en cada uno de estos tres mapas de características corresponde a una región 8x8 de la imagen de entrada. Siguiendo con la arquitectura de la red, es importante destacar que emplean capas convolucionales deformables, pues tienen en cuenta que el comportamiento de una imagen de lente *fisheye* es diferente dependiendo de la zona, como ya se ha comentado anteriormente.

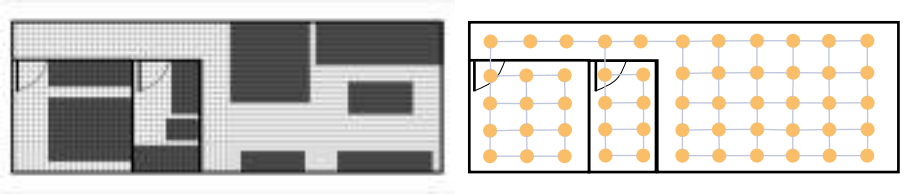
Durante el entrenamiento, se utiliza toda la imagen y, además, se trata de un aprendizaje autosupervisado. Para esto último, el algoritmo usado para el entrenamiento utiliza la matriz de homografía para generar pares de imágenes desde distintos puntos de vista. Sin embargo, como ya se ha comentado antes, este tipo de transformación no se puede aplicar directamente sobre la imagen de lente *fisheye* debido a su distorsión. De este modo, obtienen los pares de imágenes de lente *fisheye* transformadas tras llevar a cabo los siguientes tres pasos. Primero se elimina la distorsión de la imagen original. Segundo, se le aplica una matriz de homografía. Por último, se transforma a imagen de lente *fisheye*.

2.5 Creación de mapas

Como se describirá en el apartado 2.6, algunas técnicas para resolver el problema de localización requieren un modelo del entorno para localizarse con respecto a este. Además, los mapas pueden acotar el error que se comete durante la localización. Los dos tipos de mapas más comunes en la robótica móvil son los mapas métricos (figura 2.8(b)) y los mapas topológicos (figura 2.8(c)).



(a) Mapa real



(b) Rejilla 2D de ocupación (c) Mapa topológico

Figura 2.8: Los tipos de mapas más empleados en la robótica móvil. En (a) se muestra como es el entorno real, mientras que en (b) el mapa de rejilla de ocupación (mapa 2D métrico) y en (c) el mapa topológico de ese mismo entorno.

En cuanto a los mapas métricos, se caracterizan por representar el entorno con información métrica como tamaños, distancias o medidas. Este tipo de mapas puede ser una representación bidimensional (2D) o tridimensional (3D). En el caso bidimensional, se pueden dividir en mapas basados en grafos de vistas, cuyos nodos son posiciones conocidas del entorno junto con la información capturada por los sensores desde dicha posición o mapas de rejilla de ocupación, los cuales dividen el entorno en celdas que codifican la probabilidad de ocupación de cada una con el objetivo de distinguir las áreas transitables de las que no lo son. Este tipo de mapas ayudan a la planificación de rutas y a prevenir posibles colisiones. Melo et al. [211] describen un enfoque para la generación de este tipo de mapas. Dicho método utiliza información de un sistema de SLAM (la estimación de la pose y el mapa disperso) así como una técnica de segmentación de imágenes para segmentar el suelo.

En cuanto al mapa tridimensional, este puede estar compuesto por un conjunto de puntos dispersos o por un mapa denso. Con respecto al primero, estos puntos corresponden a un conjunto de características discriminatorias presentes en el entorno, los cuales pueden haberse obtenido mediante *Structure from Motion*. En cuanto al mapa denso, este consiste en un modelo de alta resolución que puede estar constituido por nubes de puntos (a partir de cámaras estéreo y RGB-D) o polígonos. Se puede encontrar más información acerca de lo mencionado en este párrafo en la sección V en [62].

Por el contrario, los mapas topológicos representan el entorno de una forma más

abstracta, modelando la geometría mediante un grafo compuesto por nodos (que representan lugares distintivos del entorno) y enlaces (que definen las relaciones de conectividad entre los primeros). Luo y Shih [212] crean un mapa topológico con una distancia entre nodos constante ya que se crea un nodo cada metro. En cada nodo, la información que guardan es la imagen capturada en esa posición y las coordenadas basadas en la odometría. Este mapa lo usan posteriormente para determinar si el robot ha alcanzado la posición destino, comparando imágenes con redes neuronales. Garcia-Fidalgo y Ortiz [213] generan un mapa topológico donde cada nodo representa un conjunto de imágenes similares definidas por un descriptor global promedio y un índice de características binarias basado en un enfoque de bolsa de palabras. A su vez, cada imagen se representa por un descriptor global y un conjunto de características locales. Se puede encontrar un estudio exhaustivo de varios trabajos en los que se crea un mapa topológico para resolver Visual-SLAM en [214]. Además, estos son clasificados en función del tipo de descriptor: global, local, BoVW (acrónimo del inglés *Bag of Visual Words*) y combinación de estos.

Comparando ambos tipos de mapas, los mapas topológicos suelen ser menos complejos y requieren de menos memoria que los métricos. Por lo tanto, tienden a simplificar y a hacer que sean más rápidas determinadas tareas como la localización o la planificación de trayectorias. A pesar de esto, los mapas métricos son la solución más adecuada para la navegación, ya que, con estos, a diferencia de los topológicos y debido a su precisión, es posible evitar obstáculos. Por lo tanto, el tipo de mapa se debe escoger en función de la tarea que se quiera realizar, pero también dependiendo de si se trata de un entorno de gran escala, en cuyo caso la solución más adecuada es el topológico al ser más robusto. Teniendo en cuenta que estos dos tipos de representación del entorno tienen fortalezas y debilidades complementarias, con objeto de aprovechar las ventajas que tienen los mapas topológicos y métricos por sí solos, surgen los mapas híbridos que combinan ambos tipos de información. Por ejemplo, Badino et al. [215] integran datos métricos directamente en los nodos topológicos de modo que contengan información de su ubicación métrica real. Además, crean una base de datos de información visual compuesta por características locales junto con su correspondiente referencia al nodo con la ubicación desde la que se observó. Para crear este mapa, se utilizaron los datos adquiridos por un vehículo equipado con cámaras y GPS mientras seguía una trayectoria.

En este misma línea, se puede utilizar un enfoque jerárquico para combinar la información como ocurre en los mapas jerárquicos. En este caso, el mapa del entorno se compone de varias capas con diferentes niveles de granularidad. Ozkil et al. [216] crean un mapa híbrido jerárquico de dos niveles. En el inferior, se encuentran los mapas métricos que son rejillas de ocupación 2D y representan regiones locales. Además, a los mapas métricos les añaden información sobre nombre de lugares de interés y, donde se superponen dos o más mapas, etiquetas visuales que actúan como *landmarks* artificiales. En el superior, un mapa topológico representa la interconectividad de estas regiones. Payá et al. [217] proponen un mapa topológico jerárquico del entorno compuesto por tres niveles y con información visual proporcionada por un sistema de visión omnidireccional. El mapa de bajo nivel representa las imágenes capturadas y las relaciones topológicas que existen entre ellas. En el nivel intermedio, se representan,

mediante imágenes representativas, grupos de imágenes cercanas espacialmente. Por último, el mapa de alto nivel se compone de las diferentes estancias y las relaciones de conectividad entre ellas.

En los últimos años, además de representar el entorno como se ha descrito en este apartado, se está complementando esta información espacial con información semántica. Mientras que el primer tipo de información es suficiente para que el robot sea capaz de localizarse o planificar su trayectoria, la información semántica consigue facilitar ciertas tareas que son más complejas (esto es, las que no solo consisten en ir desde un punto a otro evitando obstáculos) y requieren que el robot sea capaz de interpretar su entorno de forma cognitiva. Esto también es muy necesario en aquellos robots que realicen una interacción con los humanos, ya que nosotros no definimos nuestro entorno como coordenadas (mapas métricos) o nodos (mapas topológicos). De este modo, proporcionar al robot de información semántica hace que este entienda el entorno tal y como lo hacemos nosotros, mejorando así la interacción robot-humano. Por ejemplo, Qi et al. [218] proponen un método de creación de mapas de rejilla con información semántica para la navegación de robots en entornos domésticos. Para los experimentos, utilizan un robot FABO que dispone de un SONAR y visión estéreo. Con el fin de evaluar el método propuesto, un usuario ordenó al robot a ir a seis lugares (como mueble de televisión, escritorio, sofá o cama) empleando un lenguaje natural.

2.6 Localización visual

El robot móvil debe resolver la tarea de localización y ha de realizarla de forma suficientemente precisa para poder conseguir una navegación autónoma que sea eficiente, pero también segura. Como ya se ha comentado anteriormente, se trata de uno de los módulos clave en la navegación de un robot móvil, ya que, por ejemplo, no podrá planificar su trayectoria si no sabe dónde se encuentra. Por dicho motivo, aunque se trata de una línea de investigación en la que se han realizado numerosos trabajos, aún sigue siendo objeto de estudio, pues se busca mejorar la precisión y autonomía del robot en entornos extensos, complejos, cambiantes y a largo plazo.

En la figura 2.9, se muestra un esquema con dos formulaciones para resolver la localización como son *image retrieval* y la odometría visual. Ambas técnicas se describen con mayor detalle a continuación.

2.6.1 Técnica *Image retrieval*

En la literatura podemos encontrar varios trabajos en los que se formula la localización como un problema de recuperación de imágenes. De hecho, se asemeja mucho a como los humanos nos localizamos en un entorno ya visitado anteriormente.

Este proceso se divide principalmente en dos etapas, una *offline* y otra *online*. En la primera etapa, se genera un modelo visual del entorno, el cual suele ser un conjunto de descriptores (obtenidos a partir de un conjunto de imágenes representativas del entorno). Además de información visual, este mapa puede contener información espacial acerca de dónde se capturaron las imágenes con las que se ha creado el modelo visual. En cuanto a la creación del modelo visual, los descriptores pueden ser de apariencia

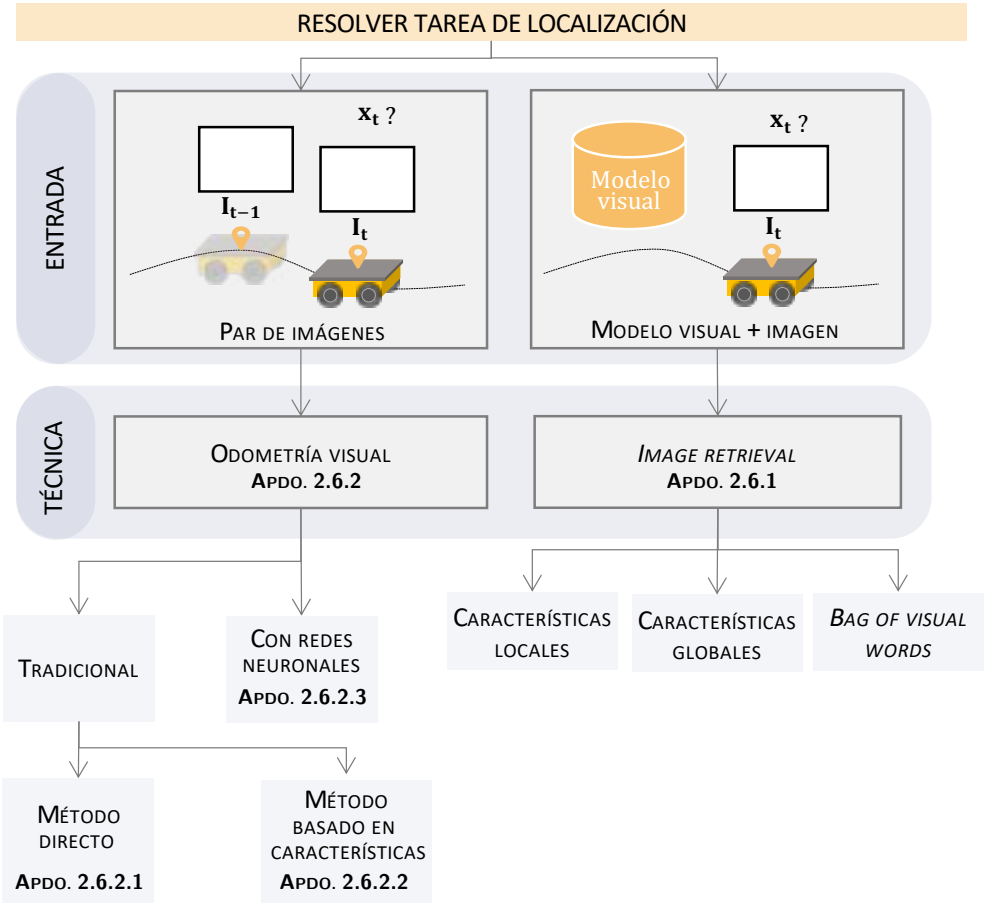


Figura 2.9: Técnicas y sus respectivos métodos para resolver la tarea de localización en función de la información que se tiene.

global [219], de características locales o un histograma de frecuencia de bolsa de palabras visuales. Por ejemplo, Román et al. [220] extraen descriptores de apariencia global, como son HOG y Gist, mientras que Zhang et al. [221] generan una bolsa de palabras visuales basada en características SURF.

En la segunda etapa, se captura una imagen desde la posición actual (imagen de test) y se extrae su correspondiente descriptor o descriptores. Empleando el modelo visual creado en la etapa anterior, se realiza una búsqueda, habitualmente mediante un algoritmo de *k-nearest neighbors* (k-nn), de los k descriptores más similares, correspondiendo cada uno de ellos a una imagen. Cada una de las k imágenes recuperadas representan una posición candidata. La pose actual se puede estimar mediante una interpolación ponderada de las poses de dichas k imágenes.

En el caso de que únicamente se recupere la imagen más similar $k = 1$, se determina que la imagen de test se capturó en la misma posición que la imagen recuperada. Cabrera et al. [170] proponen utilizar una red neuronal convolucional para realizar dos tareas. La primera función es determinar en qué estancia es más probable que se encuentre el robot, es decir, tratando la localización como un problema de clasificación. La segunda función de la red neuronal convolucional, la cual ha sido entrenada para

la primera tarea, es proporcionar un descriptor de apariencia global a partir de una de sus capas intermedias. Por lo tanto, a la vez que la red determina la estancia estimada se extrae el descriptor de la imagen de entrada y se obtiene el descriptor más similar que se encuentra en dicha estancia. La posición en la que fue capturada la imagen correspondiente a dicho descriptor es la considerada como la posición estimada para la imagen de test.

Sin embargo, para ciertas tareas se requiere una localización bastante precisa. Para ello, otro procedimiento consiste en estimar la pose relativa entre las imágenes recuperadas y la de consulta mediante correspondencias 2D-2D [221, 222] o correspondencias 2D-3D. En este último tipo de correspondencias, el conjunto de puntos 3D pueden corresponder a un mapa global de la escena o un mapa local cuyos puntos 3D se han generado a partir de las imágenes recuperadas. Como ejemplo de esto último, Jiang et al. [223], una vez recuperadas las k imágenes candidatas, las agrupan de forma que las cámaras de un mismo *cluster* sean capaces de capturar los mismos puntos 3D. Esto lo realizan debido a que, como emplean descriptores globales para la recuperación, puede darse el caso de que las imágenes recuperadas sean similares en cuanto a características globales, pero hayan sido capturadas desde diferentes ubicaciones. Para finalizar, se realiza una búsqueda de coincidencias entre los puntos 2D detectados en la imagen de consulta y los puntos 3D de cada *cluster*. La estimación de la pose se realiza resolviendo el problema de PnP (siglas del inglés *Perspective* N-*Points*) con RANSAC para cada uno de estos *clusters*, siendo el *cluster* que presente mayor número de *inliers* el que determinará la pose de la cámara para la imagen de consulta.

Por otro lado, Zhou et al. [222] describen un esquema en el que, tras recuperar las k imágenes más similares utilizando descriptores DenseVLAD [224], se calculan las poses relativas entre las imágenes recuperadas y la de test a través de la matriz esencial. Finalmente, se obtiene la pose absoluta a partir de las poses absolutas conocidas y de las matrices esenciales. Para la estimación de la pose relativa, proponen tres enfoques: uno convencional basado en características SIFT y la solución de 5 puntos [225], otro basado en una red neuronal convolucional para obtener la regresión directa de una matriz esencial y un último enfoque híbrido en el que tanto la descripción de características como la búsqueda de correspondencias 2D-2D se lleva a cabo mediante redes neuronales, pero la matriz esencial se estima mediante la solución de 5 puntos.

En [226], Humenberger et al. realizan varios experimentos exhaustivos con el fin de analizar el comportamiento de la recuperación de imágenes para resolver la localización visual mediante tres modelos: (1) aproximación de la pose, (2) sin un mapa 3D global y (3) con un mapa 3D global.

Sin embargo, este enfoque no es factible cuando el robot se mueve por un entorno desconocido, pues no se dispone de ese mapa visual. En otras palabras, para poder resolver el problema de localización mediante recuperación de imágenes, el robot debe haber recorrido previamente dicho entorno o disponer de una base de datos del mismo como Google Street View en el caso de realizarse en entorno exteriores. Esta base de datos es la que utilizan Fernández et al. [227] para realizar una evaluación comparativa de diferentes métodos de descripción de características visuales para resolver la localización en exteriores.

2.6.2 Técnica de odometría visual

El proceso conocido como odometría estima la posición y la orientación con respecto a la localización inicial del robot móvil a partir de los datos obtenidos de los sensores dispuestos a bordo del mismo. Los sensores más empleados para dicho objetivo son los *encoders*, RADAR, IMUs, LiDAR y cámara. Mohamed et al. [228] analizan las diferentes técnicas de odometría autónoma agrupándolas en cinco categorías que vienen definidas por el tipo de sensor empleado. Centrándose más en el uso de visión, Alkendi et al. [229] exponen diferentes enfoques de odometría visual y de una odometría inercial visual, todos ellos propuestos durante el periodo de 2016 hasta 2021.

La odometría visual se caracteriza por estimar el movimiento a partir de las imágenes capturadas, en dos instantes de tiempo distinto, por una sola cámara (monocular [35, 230]), por dos cámaras (estéreo [231, 232]) o por múltiples cámaras [233, 234]. Los métodos de odometría visual se pueden dividir en dos grupos, en función de si se minimiza el error fotométrico (método directo [9, 232, 235]) o de proyección (método basado en características [35, 36, 233, 234]) entre los píxeles para estimar el movimiento de la cámara. En esta línea, Agostinho et al. [236] presentan una revisión de los métodos de odometría visual en la conducción autónoma.

2.6.2.1 Método directo

Por un lado, el método directo utiliza toda la información de la imagen calculando la diferencia de intensidad. La finalidad de este método es realizar un seguimiento de la cámara mediante alineación directa de las imágenes y construir mapas de profundidad densos o semi densos. Dado que no se basa en puntos característicos, este método es más robusto y eficaz en aquellos entornos que no tengan mucha textura (poca presencia de puntos distintivos). Sin embargo, se basa en información fotométrica por lo que es sensible ante un cambio de iluminación y movimientos bruscos de la cámara.

Dentro de este enfoque, las técnicas propuestas se pueden clasificar en función de si realizan una búsqueda de correspondencia basada en región o si hacen uso de algún algoritmo de flujo óptico [237].

Como ejemplo de estimación de la pose basada en la alineación directa de la imagen se encuentra LSD-SLAM [9] (acrónimo del inglés *Large-Scale Direct* monocular SLAM) propuesto por Engel et al. [9]. En este sistema de SLAM, la representación del entorno viene dada por un gráfico de poses con *frames* claves, donde cada uno se define por una imagen, un mapa de profundidad inverso y la varianza de la profundidad inversa. Toda esta información se emplea para calcular la pose relativa de una nueva imagen minimizando el error fotométrico normalizado por la varianza. Como extensiones de este último, podemos encontrar dos trabajos más en los que propone adaptar este sistema de SLAM directo a otras configuraciones del sistema de visión: estéreo [231] y cámara omnidireccional (catadióptrica o de lente *fisheye*) [238].

Liu et al. [232] también proponen un algoritmo de odometría visual directa en el que se emplean cámaras con lente *fisheye*. Además, los autores plantean que la configuración de este sistema de visión sean dos cámaras con lente *fisheye* dispuestas en configuración estéreo. Con esto último se elimina la incertidumbre de la escala.

2.6.2.2 Método basado en características

Por otro lado, el método basado en características dispone de un primer módulo en el que se extraen puntos característicos con alguna técnica de extracción y, posteriormente, se realiza un seguimiento de estos o se buscan correspondencias comparando los descriptores. Estos métodos se pueden dividir en tres grupos en función de la dimensión de las correspondencias con las que se va a estimar el movimiento: (1) 2D-2D, (2) 3D-2D o (3) 3D-3D. En [239], se describen los algoritmos para los tres casos. A continuación, se describen algunas de las técnicas propuestas empleando características. Además, en la figura 2.10 se muestra un esquema de los procedimientos en la odometría visual basada en características.

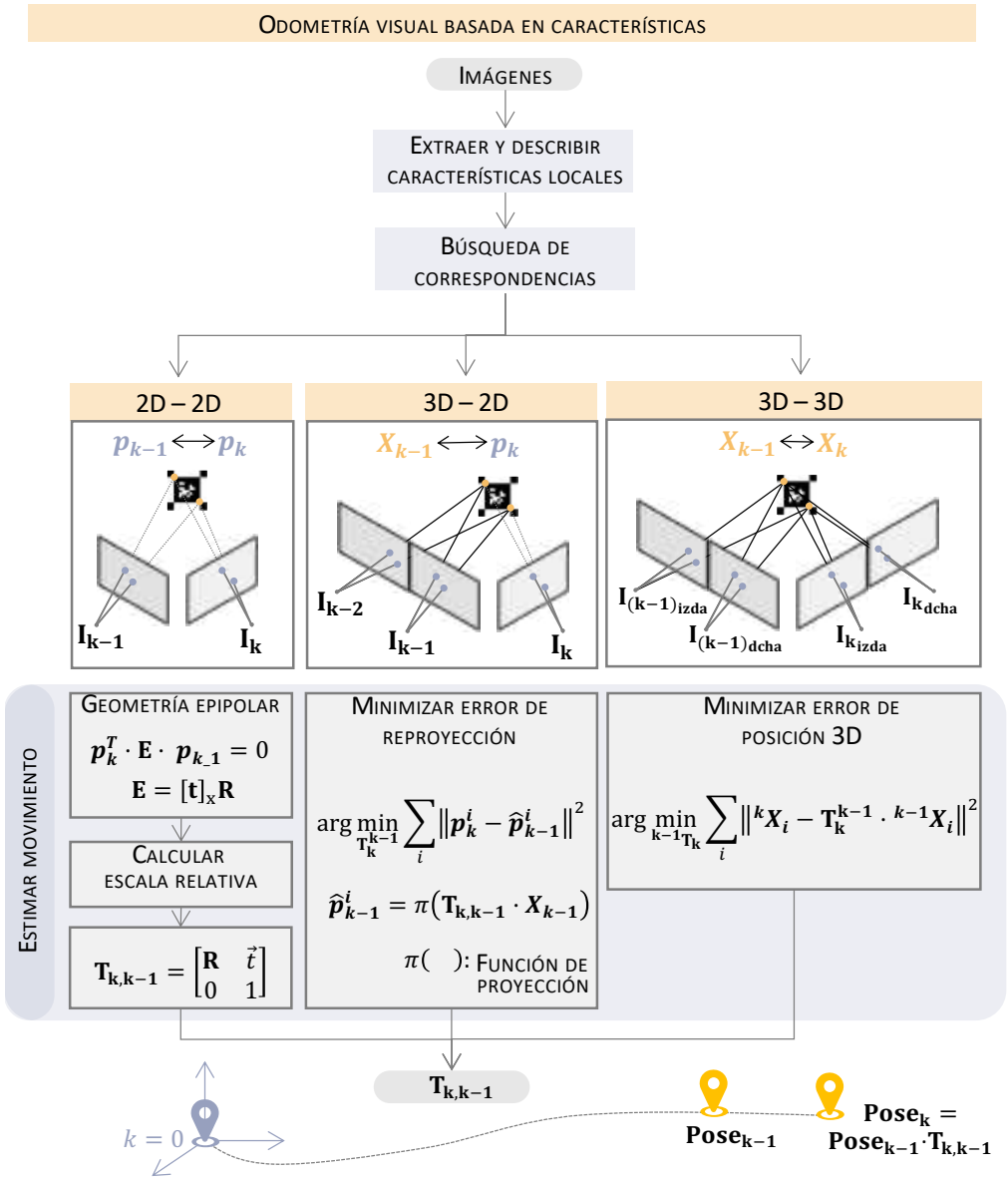


Figura 2.10: Diagrama de los principales pasos para implementar la técnica de odometría visual a partir de características locales. También se muestra cómo estimar el movimiento relativo en función de las dimensiones de los pares de correspondencias.

Mur-Artal et al. [35] proponen calcular la pose relativa de forma paralela con dos modelos: una matriz de homografía y una matriz fundamental. Ambos modelos se calculan a partir de un conjunto de correspondencias basadas en ORB. Para cada uno de estos modelos, los autores calculan una puntuación que determinará cuál de ellos finalmente se utilizará para calcular el movimiento. En el caso de que el método escogido sea la homografía, descomponen dicha matriz para obtener las ocho posibles soluciones, tal y como expone el método propuesto por Faugeras y Lustman [240]. De todas ellas, escogen aquella solución con la que, tras triangular, el mayor número de puntos se encuentren delante de ambas cámaras y, además, obtenga un menor error de reproyección. En el caso de que el método escogido sea la matriz fundamental, calculan la matriz esencial a partir de esta y escogen de las cuatro soluciones posibles aquella que sea la mejor, siguiendo la misma metodología que anteriormente con la homografía. Posteriormente, Mur-Artal y Tardos [36] adaptan el trabajo anterior para que el sistema de visión no sea únicamente monocular sino que también pueda ser estéreo o empleando cámaras RGB-D (*Red Green Blue-Depth*). Sin embargo, tanto ORB-SLAM[35] como ORB-SLAM2 [36] están diseñados para cámaras que sigan el modelo pinhole. Teniendo en cuenta esto, Campos et al. [241] sugieren extraer, en módulos diferenciados, las partes del sistema en las que se emplea el modelo de cámara. Además del modelo pinhole, se encuentra disponible el modelo de Kannala-Brandt [7] para el caso de imágenes de lente *fisheye*. Por lo tanto, en función del tipo de cámara se proporcionaría el módulo de cámara correspondiente. Este nuevo sistema propuesto se llama ORB-SLAM3, creado a partir de dos trabajos anteriores (ORB-SLAM2 [36] y Visual Inertial ORB-SLAM [242]).

Seok y Lim [233] proponen un nuevo algoritmo de odometría visual para un sistema de visión con una configuración que se caracteriza por poseer una amplia línea de base, así como un amplio campo de visión. El sistema de visión que emplean proporciona un campo de visión de 360°, ya que se compone de cuatro cámaras con un campo de visión de 220° cada una. En el algoritmo, los autores realizan una transformación a las imágenes ojo de pez mediante un modelo de proyección híbrido. La finalidad de este primer paso es asegurarse que las zonas de solape tengan la menor distorsión posible, es decir, que sean lo más perspectivas posibles. Estas imágenes rectificadas serán las que se utilicen en los siguientes pasos del algoritmo. Tras este primer paso, se lleva a cabo una extracción de características ORB. Por un lado, los autores proponen realizar un seguimiento, empleando el algoritmo de seguimiento KLT (Kanade–Lucas–Tomasi). Por otro lado, se realiza una búsqueda de coincidencias, usando el algoritmo del vecino más cercano, entre las zonas de solape de las vistas. Finalmente, la pose de la cámara es estimada utilizando P3P con RANSAC, a partir de correspondencias 2D-3D.

Javed y Kim [234] resuelven la localización con un algoritmo de odometría visual cuyas imágenes son capturadas por una cámara Ladybug. Se extraen características de cada una de las cinco imágenes proporcionadas por esta cámara y se hace un seguimiento con el algoritmo KLT. Tras eliminar las correspondencias que se hayan rastreado erróneamente, las características se proyectan sobre la esfera unidad y se usan para estimar la pose con un algoritmo de 8 puntos. Los autores evalúan este método con tres configuraciones (dos, tres y cinco cámaras), obteniendo el menor error al estimar la trayectoria con las cinco cámaras (Ladybug). También comparan

este método con dos ya conocidos (ORB-SLAM-mono y viso2-mono) consiguiendo mejores resultados.

2.6.2.3 Método con redes neuronales

Al igual que con los métodos de extracción de características, debido al desarrollo de las redes neuronales también podemos encontrar métodos que, a diferencia de los métodos tradicionales (basados en la geometría), recuperan la pose empleando redes neuronales. A continuación se describen algunos de estos, entre los que existen aquellos que se caracterizan por ser supervisados [243] y otros no supervisados [244].

Wang et al. [243] presentan DeepVO cuya arquitectura se basa en redes neuronales convolucionales recurrentes (RCNN, siglas del inglés *Recurrent Convolutional Neural Networks*). Teniendo en cuenta que se trata de un proceso *end-to-end*, es decir, estima la pose directamente a partir de una secuencia de imágenes, no es necesaria la calibración de la cámara ni ningún otro módulo de un algoritmo de odometría visual convencional. En lo que respecta a la arquitectura, el hecho de combinar redes neuronales convolucionales y redes neuronales recurrentes hace que, con las primeras, se aprendan las características efectivas que son adecuadas para el problema de odometría visual. Por su parte, a partir del segundo tipo de redes convolucionales, se modelan las dinámicas y las relaciones secuenciales.

Li et al. [244] proponen UnDeepVO el cual, teniendo como entrada un conjunto de imágenes monoculares capturadas consecutivamente, estima la pose de la cámara monocular con 6 grados de libertad y mapas de profundidad. En cuanto a la estimación de la pose, la arquitectura empleada es una red neuronal convolucional basada en VGG. Tras la última capa convolucional, hay dos grupos de capas *fully-connected*, uno para predecir la rotación (definida por los ángulos de Euler) y otro destinado a predecir la traslación. Una característica destacada de UnDeepVO es que el aprendizaje es no supervisado, las funciones de pérdida se entrenan por retropropagación ya que se basan en restricciones geométricas (no en datos etiquetados). Para minimizar los errores en el movimiento, se utiliza una función de pérdida temporal con restricciones geométricas entre dos imágenes monoculares consecutivas que comprende pérdida de consistencia fotométrica y la pérdida de registro geométrico 3D.

Pandey et al. [245] proponen una arquitectura que, tomando el flujo óptico como entrada, aprende una descripción densa de características mediante redes neuronales convolucionales que, después, es la entrada a una estructura LSTM (siglas del inglés *Long short-term memory*) bidireccional (consta de dos LSTM). Con esto último, el objetivo es modelar las relaciones temporales entre las salidas de las últimas capas de las redes neuronales convolucionales en dimensiones superiores.

3 Cámara Garmin VIRB 360 y métodos de calibración

El empleo de cámaras omnidireccionales como sensor para determinadas aplicaciones ha aumentado en los últimos años. Como ya se ha comentado en el capítulo anterior, esto se debe a que estos sistemas de visión son capaces de proporcionar una mayor cantidad de información del entorno en una única imagen o disparo. Dentro de este tipo de sistemas de visión, hay varias configuraciones (ver apartado 2.3.1). Este capítulo se centra en la configuración escogida para este trabajo que corresponde a la compuesta por dos sensores opuestos equipado cada uno de ellos con una lente *fisheye*.

En este capítulo se describirán dos herramientas *Open Access* para calibrar cámaras con lentes *fisheye* y, posteriormente, se emplearán para estimar los parámetros intrínsecos de la cámara Garmin VIRB 360.

3.1 Introducción

En los últimos años, las cámaras (o sistemas de visión) omnidireccionales se han utilizado en numerosas aplicaciones ya que, como consecuencia del campo de visión que poseen, proporcionan una visión de la escena mayor que las cámaras convencionales. Las diferentes configuraciones que encontramos son: una cámara que gira, varias cámaras apuntando hacia diferentes direcciones con cierto solapamiento en sus campos de visión o combinando una cámara convencional con elementos refractores (lentes) de gran angular, como la lente *fisheye*, o con elementos refractores y superficies reflectantes (espejos). A esta última configuración se le conoce como cámara catadióptrica. Por lo que se refiere a las principales características de las lentes *fisheye*, estas son su corta distancia focal y su amplio campo de visión, el cual puede incluso superar los $180°$.

Por otro lado, cabe señalar que las cámaras omnidireccionales se pueden clasificar

en dos categorías: centrales y no centrales. Las cámaras que pertenecen a la primera categoría satisfacen la propiedad del punto de vista (o centro de proyección) único, es decir, todos los rayos se cruzan en solo un punto 3D. Esta propiedad también la cumplen las cámaras convencionales, pues los rayos ópticos se cruzan en el llamado centro óptico de la cámara.

En relación con lo anterior, cabe destacar que la restricción del punto de vista único aporta varios beneficios. Por un lado, facilita la recuperación de la dirección del rayo de un determinado píxel, siempre que la cámara se encuentre calibrada. Por otro lado, permite que los píxeles de la imagen se puedan proyectar sobre una esfera cuyo centro coincide con el centro de proyección único. Además, el hecho de que la cámara omnidireccional cumpla esta restricción hace que se pueda aplicar la geometría epipolar, la cual solamente es válida con cámaras centrales.

En cuanto a los sistemas catadióptricos centrales, se encuentran los espejos hiperbólicos y elípticos con una cámara estándar (pinhole) y el espejo parabólico con una cámara ortográfica, tal y como demuestran Baker y Nayar [246]. En principio, las cámaras *fisheye* no tienen un único punto de vista efectivo único, sino que presentan un locus de centros de proyección al cual se le llama diacústico [247]. No obstante, en la práctica, este locus puede aproximarse a un único punto de vista, por lo que es habitual asumir que las cámaras *fisheye* son centrales [248].

Las cámaras con un gran campo de visión debido a la combinación de una cámara convencional con un espejo (catadióptricas) o una lente *fisheye* tienen la ventaja de capturar más información del entorno, pero a cambio la imagen que se obtiene presenta una mayor distorsión. En cuanto a la calibración se refiere, el inconveniente que presentan las cámaras *fisheye* y las catadióptricas es que no se pueden calibrar con método convencionales (proyección en perspectiva o modelo pinhole), de tal modo que se han propuesto un número considerable de modelos para aproximar el proceso de proyección de dichas cámaras. Puig et al. [249] presentan una clasificación y comparación de varios de los métodos de calibración para sistemas omnidireccionales, concretamente para los dos tipos más empleados, los catadióptricos y con lente *fisheye*.

Todos los modelos de cámara dependen de ciertos parámetros que deben estimarse realizando un proceso de calibración. Sin embargo, en este proceso no solo se estiman los parámetros intrínsecos sino también los extrínsecos, que corresponden a la orientación y posición. Existen varios procedimientos para la calibración [249], como autocalibración [250], calibración basada en líneas [251], calibración basada en puntos 3D [252], pero uno de los más tradicionales consiste en obtener un conjunto de puntos característicos en la imagen, que suelen ser las esquinas de un patrón de calibración, que se empleará para estimar los parámetros de calibración (intrínsecos y extrínsecos) que minimicen el error al reproyectar dicho conjunto de puntos con el modelo de cámara.

A este respecto, el proceso de calibración es crucial para muchas aplicaciones, particularmente aquellas que requieran información métrica del entorno. Asimismo, si no se consigue una calibración precisa, esto puede incorporar error en las aplicaciones en las que se emplee. Además, en el caso de las cámaras con lente *fisheye*, los modelos geométricos no solo se usan para relacionar el entorno con la imagen, sino que también se utilizan para convertir una imagen *fisheye* en una imagen en perspectiva

(rectificación). En relación con esto último, Fan et al. [253] analizan exhaustivamente el progreso en la rectificación de imágenes de gran angular, centrándose tanto en los modelos de cámara y los modelos de distorsión, como en los dos tipos de métodos de rectificación de imágenes: métodos tradicionales basados en geometría y métodos basados en aprendizaje profundo.

Teniendo en cuenta que la cámara empleada en este trabajo se compone de dos lentes *fisheye*, a continuación, se detallarán algunos de los modelos aptos para este tipo de cámara. En esta misma línea, Kumar et al. [254] presentan los modelos de cámaras *fisheye* más populares dividiéndolos, principalmente en modelos geométricos clásicos (proyección equidistante, estereográfica u ortográfica), modelos algebraicos (polinomiales o de división) y modelos esféricos (modelo de campo de visión, de cámara unificada, de cámara unificada mejorada y de doble esfera).

En la categoría de modelos esféricos, se encuentra el modelo de cámara unificada (UCM, acrónimo del inglés *Unified Camera Model*) para cámaras catadióptricas centrales (espejo hiperbólico, parabólico o elíptico) propuesto inicialmente por Geyer y Daniilidis [255] y mejorado posteriormente por Barreto y Araujo [256]. Este modelo consta de una primera proyección sobre una esfera unitaria virtual, seguida de otra proyección en perspectiva sobre el plano imagen cuyo centro de proyección se encuentra a una distancia ξ del centro de la esfera. Como se ha especificado al principio, este modelo fue propuesto para las cámaras catadióptricas centrales. Pese a ello, en varios trabajos [247, 257, 258], se demostró que también podía aplicarse a cámaras *fisheye*. Mei y Rives [16] presentan una versión ligeramente modificada de este modelo y un método de calibración empleando un patrón de calibración. Es importante mencionar que los autores justifican que puede emplearse tanto con cámaras catadióptricas centrales como con cámaras *fisheye* o esféricas. Posteriormente, se sugirieron dos modelos más que se basan en este. En 2016, Khomutenko et al. [4] proponen que la superficie de proyección sea un elipsoide. A este método se le conoce como modelo de cámara unificada extendida (EUCM, acrónimo del inglés *Extended Unified Camera Model*). Posteriormente, en 2018, Usenko et al. [3] sugieren otra modificación del método unificado en la que el punto 3D se proyecta primero sobre una esfera y después sobre otra esfera, de ahí que se le denomine modelo de cámara de doble esfera (DSCM, acrónimo del inglés *Double Sphere Camera Model*).

En cuanto a modelos basados en polinomios, se encuentra el propuesto por Kannala y Brandt (KB) [7] donde la proyección sobre el plano imagen viene dado por un polinomio que depende del ángulo de incidencia. Este polinomio se compone únicamente de exponentes impares y con un orden igual o superior a cinco. En esta misma línea, Scaramuzza et al. [10] suponen que la función de proyección a imagen, para sistemas centrales, puede describirse mediante una expansión en serie de Taylor. En este caso, el polinomio depende de la distancia al centro de la imagen y no tiene término de primer orden debido a que la primera derivada del polinomio debe ser nula cuando la distancia al centro es cero.

Gran parte de los modelos para las cámaras *fisheye* que existen en la literatura, como los que se han mencionado, asumen que la propiedad del punto de vista único se cumple. Sin embargo, también se proponen modelos en los que se considera que este tipo de cámara es no central. Por ejemplo, Tezaur et al. [259] presentan un modelo

no central que se basa en el modelo central propuesto por Scaramuzza et al. [10] pero realizando ciertas modificaciones para incorporar la caracterización del cambio de punto de vista presente en la lente *fisheye*.

Al igual que ocurre en otros ámbitos, debido al desarrollo de la inteligencia artificial, podemos encontrar algunos trabajos que proponen el empleo de redes neuronales para la calibración.

Bogdan et al. [260] presentan una red neuronal para estimar, de forma totalmente automática, los parámetros intrínsecos de la cámara a partir de una única imagen. Los parámetros intrínsecos estimados son la distancia focal (f) y el parámetro ξ, ambos correspondientes al modelo de cámara unificado. Para entrenar la red neuronal, los autores crean un conjunto de imágenes sintéticas a partir de un *dataset* compuesto por imágenes con un campo de visión de $360°$. Para la generación del conjunto de datos de entrenamiento, cada vista completa del *dataset* comentado es proyectada sobre la esfera y después se realiza una proyección sobre el plano imagen aplicando el modelo de cámara unificada asignando distintos valores para los dos parámetros (f y ξ).

En esta misma línea, Wakai y Yamashita [261] sugieren utilizar redes neuronales para predecir tanto los parámetros extrínsecos como los intrínsecos de una cámara *fisheye*. Los parámetros extrínsecos estimados son dos, ángulo de inclinación y balanceo, mientras que solo hay uno como parámetro intrínseco, la distancia focal. Además, en línea con esto último, la calibración que proponen se basa en una función trigonométrica de proyección de *fisheye*. En cuanto a la arquitectura, primero se extraen mapas de características con DenseNet preentrenada, de los cuales se conseguirá un vector de características aplicando la agrupación promedio global. Finalmente, la arquitectura se compone de tres regresores individuales (capa *fully-connected* de 2208 canales + ReLU seguida de capa *fully-connected* de 256 canales con activación mediante sigmoide).

En este capítulo se utilizan, para calibrar la Garmin VIRB 360, dos herramientas *OpenSource*, como son: calibración Basalt [38] y OCamCalib (*Omnidirectional Camera Calibration*) [10]. Como se verá en el apartado 3.3, Basalt dispone de cuatro modelos de cámaras (UCM, EUCM, DSCM y KB) para que el usuario escoja cuál quiere emplear. Respecto a la herramienta OCamCalib, esta únicamente tiene implementado un modelo de cámara y es el propuesto por Scaramuzza et al. [10].

3.2 Cámara Gamin VIRB 360

El sistema de visión en el que se centra la presente tesis es la cámara Garmin VIRB 360 [262], la cual se compone de dos lentes *fisheye* y dos sensores CMOS retroiluminados ($1/2.3''$). Cada lente tiene un campo de visión de $201.8°$ por lo que, debido a esto último y a como están posicionadas (espalda contra espalda), la cámara Garmin VIRB 360 captura un campo de visión de 360 grados, vertical y horizontal, es decir, una esfera completa. En cuanto a sus características técnicas, las más importantes se encuentran en la tabla 3.1. Además, esta cámara permite configurar el modo de óptica con cuatro opciones: 360, solo delantera, solo trasera o RAW. Cada uno de estos modos se encuentran descritos en la tabla 3.2, así como en la figura 3.2, que muestra el tipo de imagen que se obtiene con cada uno de ellos.

Figura 3.1: Cámara Garmin VIRB 360.

Tabla 3.1: Principales características técnicas de la cámara Garmin VIRB 360.

Características físicas	
Peso:	160g (con batería)
Tamaño:	39.0mm(H)×59.3mm(W)×69.8mm(D)
Características ópticas	
Sensor:	1/2.3" CMOS con iluminación trasera (2 sensores)
Número de lentes:	2
Campo de visión:	201.8 grados (cada lente)
Distancia focal efectiva	1.036mm

Para algunas tareas en visión por computador es necesario conocer la relación entre los puntos 3D del entorno y sus proyecciones en el plano imagen. Entre ellas, se encuentran la geometría epipolar y la rectificación de las imágenes *fisheye*, que son dos cuestiones de gran relevancia para esta tesis. Por ello, en este capítulo se van a mostrar algunos modelos adecuados para imágenes *fisheye* y además se van a emplear para calibrar la cámara Garmin VIRB 360.

Con la finalidad de realizar dicho proceso de calibración, se emplean dos *toolbox* dedicadas a este tipo de lente: (I) Basalt [38] (apartado 3.3) y (II) OCamCalib [10] (apartado 3.4). Como se verá, mientras que la primera tiene implementadas varios modelos de cámara, la segunda únicamente se basa en el propuesto por su autor Scaramuzza et al. [10] . Otra diferencia radica en el tipo de patrón, pues si bien la de Basalt utiliza AprilTags como patrón de calibración (figura 3.3(a)), la *toolbox* de Scaramuzza se basa en el conocido patrón de ajedrez (figura 3.3(b)).

3.3 Calibración de la Garmin VIRB 360 con Basalt

En este apartado se describe la *toolbox* de calibración Basalt, la cual se basa en marcas AprilTag (ver figura 3.3(a)) y cuya interfaz se muestra en la figura 3.4. Las características más interesantes de esta herramienta para el presente trabajo son que incluye varios modelos de cámaras, los cuales se describen en [3], y que además proporciona la matriz de transformación entre dos cámaras.

El procedimiento seguido para la adquisición de los datos es el siguiente. En primer

(a) (b)

(c)

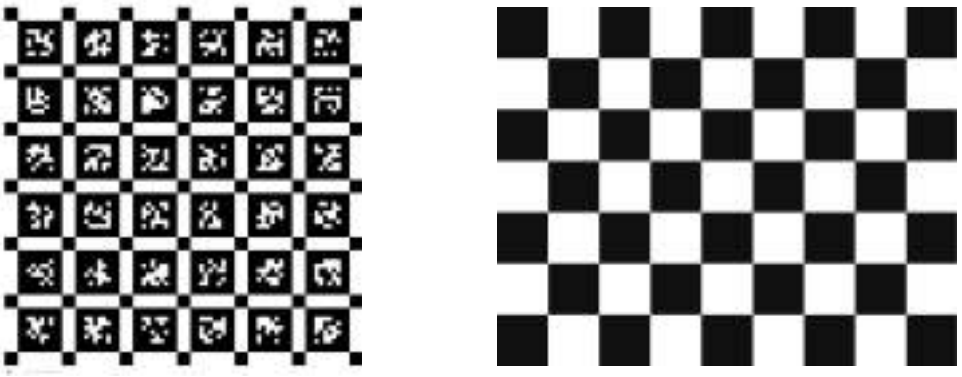
(d) (e)

Figura 3.2: Tipos de imágenes que captura la Garmin VIRB 360 en función del modo (ver tabla 3.2) configurado: par de imágenes *fisheye* (a) delantera y (b) trasera adquiridas simultáneamente con el modo RAW; (c) vista completa que proporciona al establecer el modo 360; (d) imagen en perspectiva capturada con (d) la cámara delantera y (e) la trasera.

lugar, se colocó la Garmin VIRB 360 en un trípode y se configuró para grabar vídeo seleccionando como modo óptico la opción RAW. Una vez preparado todo, se inició la

Tabla 3.2: Los modos ópticos que se pueden configurar en la cámara Garmin VIRB 360 para la captura de fotografías y vídeos.

Modo	Descripción	Resolución fotografía	Resolución vídeo
360:	Este modo genera una vista esférica completa en formato equirectangular. Ejemplo de fotografía: figura 3.2(c)	5640×2820 (15MP)	4K: 3840×2160
Solo delantera:	Este modo genera una imagen en perspectiva a partir de la fotografía capturada por la lente frontal. Ejemplo de fotografía: figura 3.2(d)	1920×1440 (3MP)	FHD: 1920×1080
Solo trasera:	Este modo genera una imagen en perspectiva a partir de la fotografía capturada por la lente trasera. Ejemplo de fotografía: figura 3.2(e)	1920×1440 (3MP)	FHD: 1920×1080
RAW:	Este modo captura una imagen sin procesar con cada lente (2 archivos). Ejemplo de fotografía adquirida con este modo: figura 3.2(a) y figura 3.2(b)	3008×3000 (2×9MP)	5.7K*: (x2) 2880×2880 5K: (x2) 2496×2496 3.5K: (x2) 1760×1760



(a) (b)

Figura 3.3: Patrón de calibración (a) AprilTag y (b) tablero de ajedrez.

grabación y se fue moviendo el tablero AprilTag (como el que se muestra en la figura 3.3(a)) al mismo tiempo que se giraba alrededor de la cámara. Del par de vídeos, se extrajo una secuencia de 560 imágenes y, además, dicho *dataset* se convirtió a formato euroc, que es uno de los dos formatos (bag y euroc) con los que trabaja dicha herramienta.

Figura 3.4: Interfaz de la herramienta de calibración Basalt.

En cuanto a los modelos de cámara empleados con esta herramienta son los siguientes: (a) el modelo de cámara unificada (UCM), dos extensiones de esta como son (b) el modelo de cámara unificada extendido (EUCM) y (c) el modelo de proyección de doble esfera (DSCM) y por último (d) el modelo de Kannala-Brandt. A continuación, se describirá cada uno de estos modelos. Cabe destacar que los parámetros de cada uno de ellos corresponden a los que se presentan en [3] ya que es donde se describen los modelos que han empleado en la herramienta Basalt. Igualmente, en cada apartado se mostrarán los valores de los parámetros intrínsecos obtenidos tras emplear dicha herramienta con dicho modelo.

3.3.1 Modelo de cámara unificada

En primer lugar, Mei y Rives [16] proponen el modelo de cámara unificada (UCM, acrónimo del inglés *Unified Camera Model*) cuya superficie de proyección es una esfera. La figura 3.5(a) muestra la proyección de un punto de la escena expresado en el sistema de referencia de la cámara ($\boldsymbol{P}$) en el plano imagen según este modelo. Dicha proyección se compone de los siguientes pasos. En primer lugar, el punto 3D ($\boldsymbol{P} = (X, Y, Z)$), se proyecta sobre la esfera unitaria ($\boldsymbol{P_e}$). Finalmente, este ($\boldsymbol{P_e}$) es proyectado en el plano de imagen ($\boldsymbol{p}$) siguiendo el modelo pinhole con un centro de proyección virtual desplazado una distancia de $\frac{\alpha}{1-\alpha}$ del centro de la esfera unitaria.

Atendiendo a lo anterior, con este modelo la normalización de las coordenadas se lleva a cabo con la siguiente función:

$$\mathcal{N}(X, Y, Z) = \frac{\alpha}{1-\alpha} + \frac{Z}{d} = \alpha \cdot d + (1-\alpha)Z \tag{3.1}$$

donde $d = \sqrt{X^2 + Y^2 + Z^2}$ y $\alpha \, \epsilon \, [0, 1]$. Así pues, la proyección del punto 3D ($\boldsymbol{P}$) con

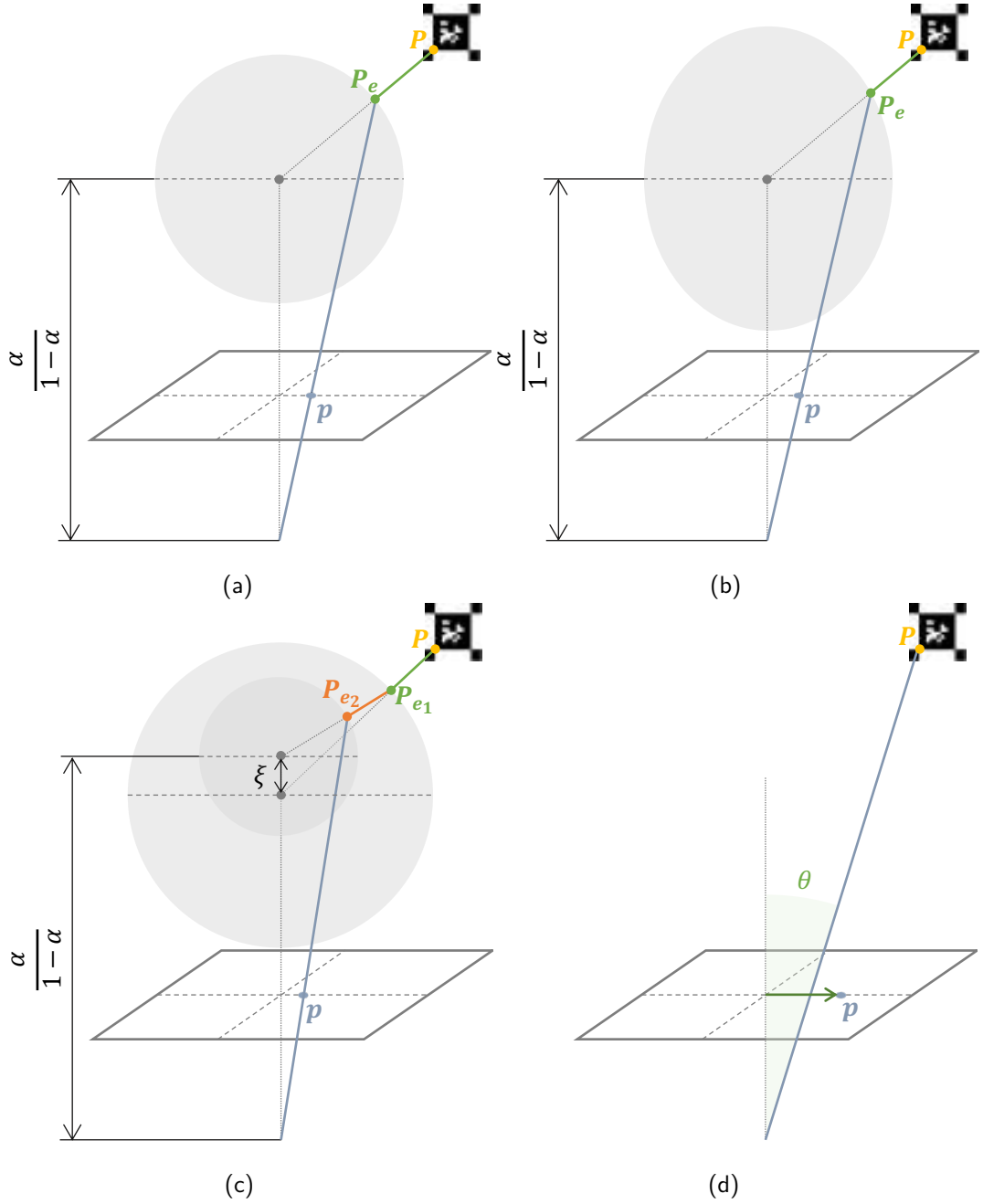


Figura 3.5: Modelos de cámara implementados en Basalt: (a) modelo de cámara unificada (UCM), (b) modelo de cámara unificada extendido (EUCM), (c) modelo de cámara doble esfera (DSCM) y (d) modelo de cámara Kannala-Brandt (KB).

coordenadas (X, Y, Z) sobre el plano imagen $(\boldsymbol{p})$ viene dado por:

$$\boldsymbol{p} = \begin{bmatrix} f_x \frac{X}{\mathcal{N}(X,Y,Z)} \\ f_y \frac{Y}{\mathcal{N}(X,Y,Z)} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} f_x \frac{X}{\alpha \cdot d + (1-\alpha) Z} \\ f_y \frac{Y}{\alpha \cdot d + (1-\alpha) Z} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{3.2}$$

siendo f_x, f_y, c_x y c_y los cuatro parámetros intrínsecos de la función de proyección del modelo pinhole. De este modo, la función de proyección de este modelo tiene cinco

parámetros intrínsecos: f_x, f_y, c_x, c_y y α. Como se ha comentado, estos parámetros son los que utilizan en [3], mientras que la formulación estándar (i. e. cómo se define en la literatura [16]) de este modelo se compone de los siguientes cinco parámetros intrínsecos: γ_x, γ_y, c_x, c_y y ξ. Existe una relación entre ambas formulaciones:

$$\xi = \frac{\alpha}{1-\alpha}, \quad \gamma_x = \frac{f_x}{1-\alpha}, \quad \gamma_y = \frac{f_y}{1-\alpha} \tag{3.3}$$

Usenko et al. [3] proponen esta nueva formulación porque con este cambio de parámetros intrínsecos se consigue una mayor estabilidad numérica de los resultados para un mismo error de reproyección. Tras comparar ambas formulaciones para el mismo modelo, observaron que la nueva formulación que proponen tiene una menor desviación estándar para los valores de los parámetros intrínsecos y, además, que el valor de la distancia focal tiene un valor cercano al de los otros modelos de cámaras que describen en ese mismo trabajo (i.e. EUCM y DSCM).

En relación con la calibración de la Garmin VIRB 360, en la tabla 3.3, se muestran los valores obtenidos para cada uno de estos parámetros tanto con la cámara delantera (cam0) como con la trasera (cam1).

Tabla 3.3: Parámetros intrínsecos del modelo de cámara UCM. Los parámetros del sensor CMOS y lente *fisheye* delantera se identifican como cam0 y cam1 corresponden a la trasera.

	f_x	f_y	c_x	c_y	α
cam0	606.981	602.101	1236.49	1250.76	0.55484
cam1	603.006	597.517	1233.4	1247.51	0.551264

3.3.2 Modelo de cámara unificado ampliado

Por otro lado, Khomutenko et al. [4] proponen un modelo de cámara unificado ampliado (EUCM, acrónimo del inglés *Extended Unified Camera Model*). Este modelo se puede interpretar como una generalización del modelo anterior, UCM, donde se modifica la superficie de proyección; en lugar de una esfera, el punto 3D se proyecta sobre un elipsoide, como se muestra en la figura 3.5(b). La diferencia con el anterior radica en la función de normalización, concretamente en d:

$$\mathcal{N}(X,Y,Z) = \alpha \cdot d + (1-\alpha)Z \tag{3.4}$$

donde ahora $d = \sqrt{\beta(X^2+Y^2)+Z^2}$. El parámetro β ajusta la forma de la superficie de proyección. Al inicio de este apartado, se ha comentado que este modelo se trata de una generalización del modelo UCM, puesto que que en el caso de que el parámetro β valga 1 la función de proyección coincidiría con la de UCM.

Considerando todo lo comentado, para este modelo, la función de proyección del punto 3D ($\boldsymbol{P}$) sobre el plano imagen ($\boldsymbol{p}$) se define como:

$$\boldsymbol{p} = \begin{bmatrix} f_x \frac{X}{\mathcal{N}(X,Y,Z)} \\ f_y \frac{Y}{\mathcal{N}(X,Y,Z)} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} f_x \frac{X}{\alpha \cdot d + (1-\alpha)Z} \\ f_y \frac{Y}{\alpha \cdot d + (1-\alpha)Z} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{3.5}$$

En definitiva, este modelo se compone de seis parámetros intrínsecos: f_x, f_y, c_x, c_y, α y β. Con este método, los valores de los parámetros intrínsecos obtenidos para la Garmin VIRB 360 son los que aparecen en la tabla 3.4.

Tabla 3.4: Parámetros intrínsecos del modelo de cámara EUCM.

	f_x	f_y	c_x	c_y	α	β
cam0	563.31	564.78	1237.2	1249.2	0.5824	0.8007
cam1	559.91	561.09	1234.7	1245.2	0.5807	0.7965

3.3.3 Modelo de cámara de doble esfera

En otro orden de cosas, Usenko et al. [3] proponen el modelo de proyección de doble esfera (DSCM, acrónimo del inglés *Double Sphere Camera Model*). Este modelo también es una ampliación del primero modelo, UCM, pues, en este caso, hay dos superficies de proyección que corresponden a dos esferas cuyos centros se encuentran desplazados una distancia ξ.

Tal y como se muestra en la figura 3.5(c), el punto 3D expresado en el sistema de coordenadas de la cámara ($\boldsymbol{P}$) se proyecta, en primer lugar, sobre la primera esfera ($\boldsymbol{P}_{e_1}$) y, en segundo lugar, sobre la segunda esfera ($\boldsymbol{P}_{e_2}$) que se encuentra desplazada una distancia ξ con respecto a la anterior. Después, se proyecta sobre el plano imagen ($\boldsymbol{p}$) mediante el modelo pinhole. Al igual que en los anteriores, en esta última proyección el centro de proyección se encuentra desplazado $\frac{\alpha}{1-\alpha}$ del centro del la segunda esfera. Según lo descrito, la normalización de las coordenadas se lleva a cabo con la siguiente ecuación:

$$\mathcal{N}(X,Y,Z) = \frac{\alpha}{1-\alpha} + \frac{\xi + Z/d_1}{d_2} = \alpha \cdot d_2 + (1-\alpha)\cdot(\xi \cdot d_1 + Z) \tag{3.6}$$

donde $d_1 = \sqrt{X^2+Y^2+Z^2}$, $d_2 = \sqrt{X^2+Y^2+(\xi\cdot d_1+Z)^2}$. La proyección del punto 3D ($\boldsymbol{P}$) con coordenadas (X,Y,Z) sobre el plano imagen ($\boldsymbol{p}$) viene dado por:

$$\boldsymbol{p} = \begin{bmatrix} f_x \frac{X}{\mathcal{N}(X,Y,Z)} \\ f_y \frac{Y}{\mathcal{N}(X,Y,Z)} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} f_x \frac{X}{\alpha\cdot d_2+(1-\alpha)\cdot(\xi\cdot d_1+Z)} \\ f_y \frac{Y}{\alpha\cdot d_2+(1-\alpha)\cdot(\xi\cdot d_1+Z)} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{3.7}$$

En resumen, este modelo de cámara tiene los siguientes seis parámetros intrínsecos: f_x, f_y, c_x, c_y, ξ y α. Tras calibrar la Garmin VIRB 360 con la herramienta Basalt seleccionando este modelo, se obtienen los siguientes resultados:

Tabla 3.5: Parámetros intrínsecos del modelo de cámara DSCM.

	f_x	f_y	c_x	c_y	ξ	α
cam0	606.94	602.00	1236.4	1250.8	2.0155e-11	0.5548
cam1	603.06	597.51	1233.5	1247.6	-9.8504e-12	0.5512

3.3.4 Kannala-Brandt

Kannala y Brandt [7] sugirieron el modelo Kannala-Brandt (KB). Se trata de un modelo genérico y se caracteriza por asumir que la distancia desde el centro óptico al punto proyectado es proporcional a un polinomio de orden n que depende del ángulo (θ) entre el punto 3D y el eje principal, como se muestra en la figura 3.5(d). En el caso de que el orden del polinomio fuese igual a 9, este viene dado por:

$$d(\theta) = \theta + k_1 \cdot \theta^3 + k_2 \cdot \theta^5 + k_3 \cdot \theta^7 + k_4 \cdot \theta^9 \tag{3.8}$$

donde k_1, k_2, k_3 y k_4 son los cuatro parámetros a estimar de este modelo. Durante la estimación se ha escogido este orden del polinomio ($n = 9$), por lo que a partir de ahora se hará referencia este modelo como KB4, para identificar que el número de parámetros que se estiman son 4.

De este modo, la proyección sobre el plano imagen ($\boldsymbol{p}$) viene dado por:

$$\boldsymbol{p} = \begin{bmatrix} f_x \cdot d(\theta) \cdot \frac{X}{r} \\ f_y \cdot d(\theta) \cdot \frac{Y}{r} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{3.9}$$

donde $r = \sqrt{X^2 + Y^2}$ y $\theta = atan2(r, Z)$.

Así pues, este modelo se compone de ocho parámetros intrínsecos: f_x, f_y, c_x, c_y, k_1, k_2, k_3 y k_4. Tras seleccionar este modelo con las imágenes capturadas con la Garmin VIRB 360, se obtuvieron los valores que se muestran en la tabla 3.6.

Tabla 3.6: Parámetros intrínsecos del modelo de cámara KB4.

	f_x	f_y	c_x	c_y	k_1	k_2	k_3	k_4
cam0	548.94	551.095	1235.4	1249.6	0.1488	-0.0307	0.00711	-1.0216e-3
cam1	548.95	550.80	1235.5	1244.5	0.1492	-0.0310	0.0071	-9.5566e-4

3.3.5 Resultados con Basalt

En los apartados anteriores se han presentado los parámetros de cada modelo y los valores obtenidos para cada caso al calibrar la cámara Garmin VIRB 360 con la herramienta de calibración Basalt. No obstante, durante el proceso de calibración la herramienta también proporciona otra serie de parámetros comunes cuyos valores se muestran en la tabla 3.7 con la finalidad de compararlos.

En la tabla 3.7 se indica, a modo de resumen, el número de parámetros que se estima para cada uno. También algunos de los parámetros de salida de la herramienta Basalt: el número de puntos detectados; el error medio de reproyección que se calcula dividiendo el sumatorio de la distancia, en píxel, entre el punto detectado y su proyectado (error de reproyección) entre el número de puntos; y la matriz de transformación entre ambas cámaras. Además, en la figura 3.6 se pueden ver los puntos detectados (color rojo) y proyectados (color magenta) para cada uno de estos modelos con la herramienta de calibración de Basalt.

Tabla 3.7: Comparativa de los cuatro modelos con la herramienta Basalt. Las cuatro últimas filas corresponden a la información de salida tras la optimización.

	UCM [16]	EUCM [4]	DSCM [3]	KB4 [7]
Número de parámetros	5	6	6	8
Número de puntos	347964	431635	260370	54884
Error de reproyección (en píxeles)	901699	360555	675752	**25547**
Error medio de reproyección (en píxeles)	2.5914	0.8353	2.5954	**0.4655**
Tiempo de optimización	34ms	39ms	13ms	7ms

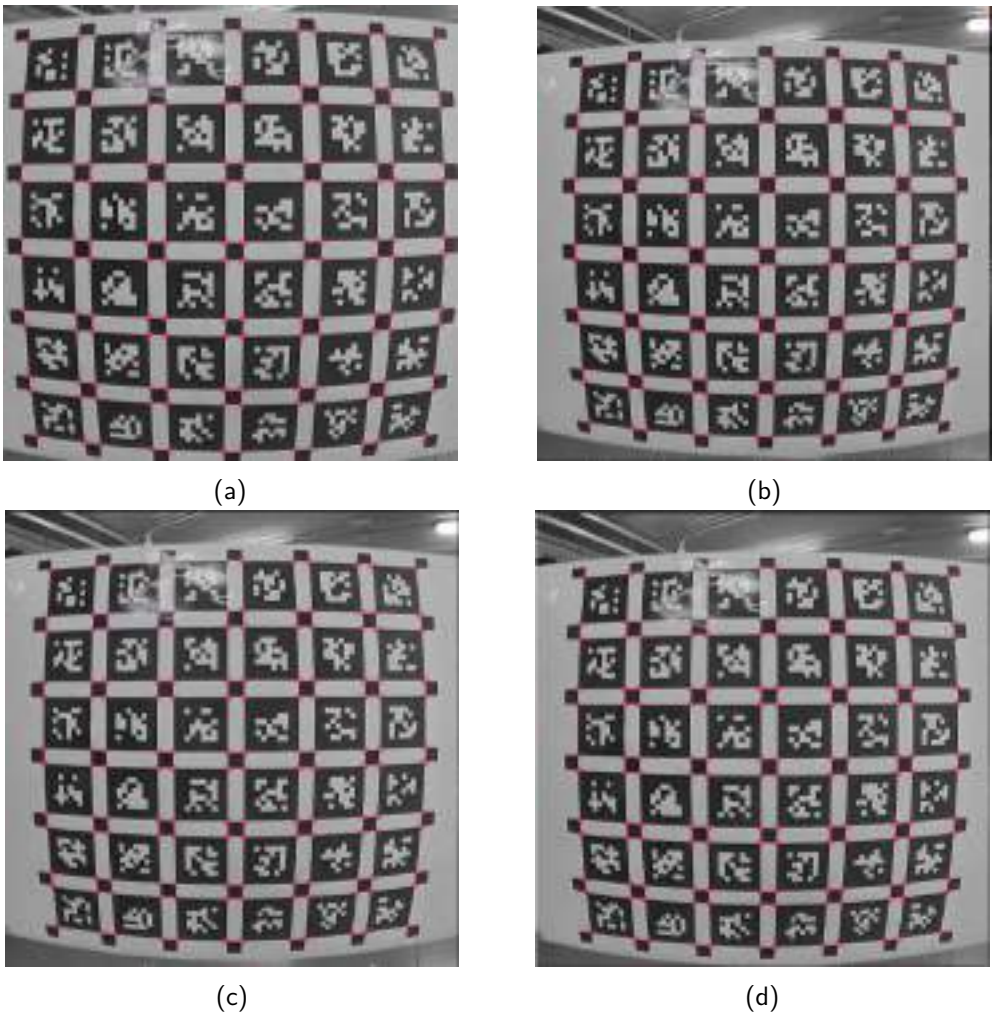


Figura 3.6: Ejemplo de imagen en la que el patrón se encuentra en el centro de una de las imágenes *fisheye*: (a) modelo de cámara unificada (UCM), (b) modelo de cámara unificada extendido (EUCM), (c) modelo de cámara doble esfera (DSCM) y (d) modelo de cámara Kannala-Brandt (KB). En rojo se muestran las esquinas detectadas y en magenta los puntos reproyectados.

Al comparar los resultados de estos parámetros, el cuarto modelo, KB4 [7], es el que menor error de reproyección tiene. Por el contrario, el primer modelo, UCM [16], junto con el tercero, DSCM [3], presentan el error de reproyección más elevado, siendo tres veces mayor que el segundo modelo, EUCM [4]. En cuanto al tiempo en la etapa de optimización del proceso de calibración, el tercer modelo, DSCM [3], es el más rápido con 13ms teniendo en cuenta el tiempo y el número de puntos que se procesan.

A pesar de que el modelo KB4 [7] presenta un error de reproyección muy bajo, tras analizar las imágenes, observamos que, con KB4 [7], en varios pares de imágenes *fisheye* los puntos no se proyectan cerca de los detectados (como se muestra en la figura 3.8). Esto se ha observado en los otros modelos, pero con menos frecuencia. Además, en el caso de KB4 [7] suele ocurrir cuando el patrón se encuentra en la zona de mayor distorsión. Por ejemplo, en la figura 3.7 se muestra el patrón de calibración en ambas imágenes y se puede ver como en el resto de modelos (ver figura 3.7(a), 3.7(b) y 3.7(c)) los puntos (magenta) se han proyectado cerca, con mayor o menor error, de las esquinas detectadas (color rojo), excepto con el modelo KB4 [7] (figura 3.7(d)).

Tabla 3.8: Transformaciones entre las dos cámaras: delantera (cam0) y trasera (cam1).

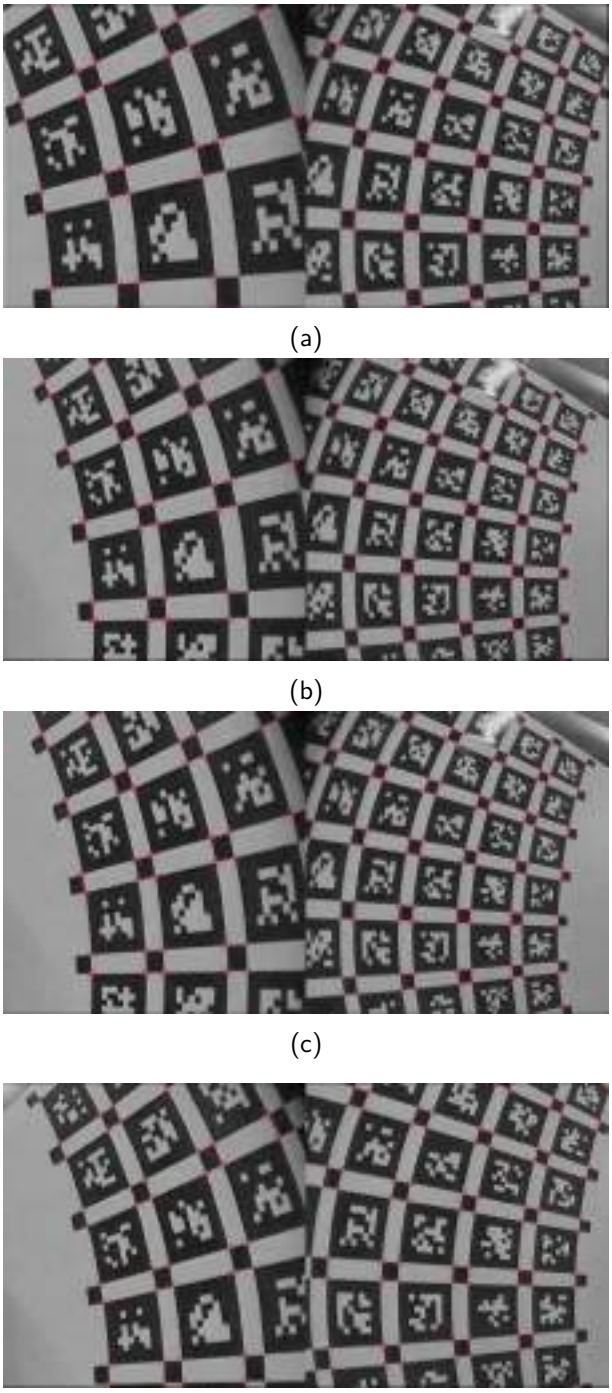
	tx	ty	tz	θ_Z	θ_Y	θ_X
UCM	$1.8921e$-04	$1.6644e$-04	-$1.4791e$-02	$179.9433°$	$0.3339°$	$179.8529°$
EUCM	-$3.6466e$-04	-$1.6957e$-05	-$2.0608e$-02	$179.9466°$	$0.1186°$	-$179.9384°$
DSCM	$1.9037e$-04	$1.6660e$-04	-$1.4756e$-02	$179.9436°$	$0.3198°$	$179.8526°$
KB4	-$1.7143e$-04	-$4.3936e$-05	-$2.1947e$-02	$179.8963°$	-$0.0107°$	-$179.9273°$

Además de los parámetros intrínsecos de cada modelo, la herramienta Basalt también estima la pose relativa entre el par de imágenes *fisheye*, cuyas matrices de transformación se muestran en la tabla 3.8. Si observamos la traslación relativa, se observa que ambas cámaras se encuentran desplazadas y que, con todos los modelos, el mayor desplazamiento se produce en el eje Z.

3.4 Calibración de la Garmin VIRB 360 con OCamCalib

En 2006, Scaramuzza et al. [10] proponen un modelo de proyección unificado para la cámara catadióptrica y *fisheye* basado en un polinomio de Taylor. Además del modelo, el cual se describe en el apartado 3.4.1, presentan la *toolbox* disponible en [263] para realizar el proceso de calibración utilizando un tablero de ajedrez.

El primer paso para poder utilizar esta herramienta para calibrar la Garmin VIRB 360 es la adquisición de las imágenes. No obstante, el procedimiento realizado para ello es distinto que el anterior. En este caso, cada lente se calibra de forma independiente. De este modo, para ello, se configuró la Garmin VIRB 360 en modo fotografía y RAW, tras lo cual se tomaron dos conjuntos de imágenes *fisheye*. Para la adquisición de ambos conjuntos, el tablero de ajedrez se mantuvo fijo mientras se movía la cámara apuntando primero con la lente delantera para la adquisición del primer conjunto y después con la trasera para el segundo conjunto. Dado que para poder calibrar el tablero debe ser capturado completamente, cada conjunto de imágenes se encuentra



(a) (b) (c) (d)

Figura 3.7: Ejemplo de imagen en la que el patrón se encuentra en la zona de solape: (a) modelo de cámara unificada (UCM), (b) modelo de cámara unificada extendido (EUCM), (c) modelo de cámara doble esfera (DSCM) y (d) modelo de cámara Kannala-Brandt (KB). En rojo se muestran las esquinas detectadas y en magenta los puntos reproyectados.

compuesto de imágenes capturadas por una misma lente. En la figura 3.10, se muestran nueve imágenes del conjunto adquirido para calibrar la lente delantera (figura 3.10(a)) y nueve imágenes del conjunto adquirido para calibrar la lente trasera (figura 3.10(b)).

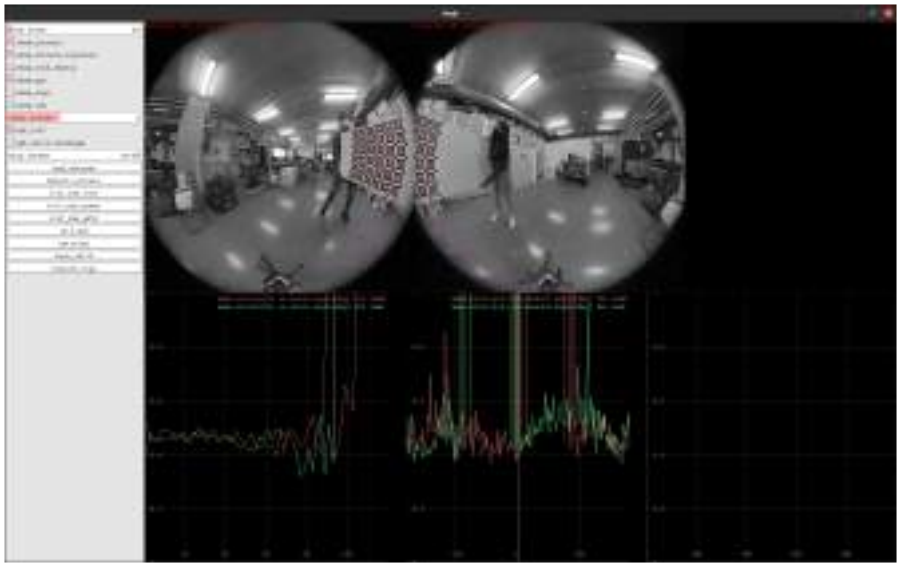
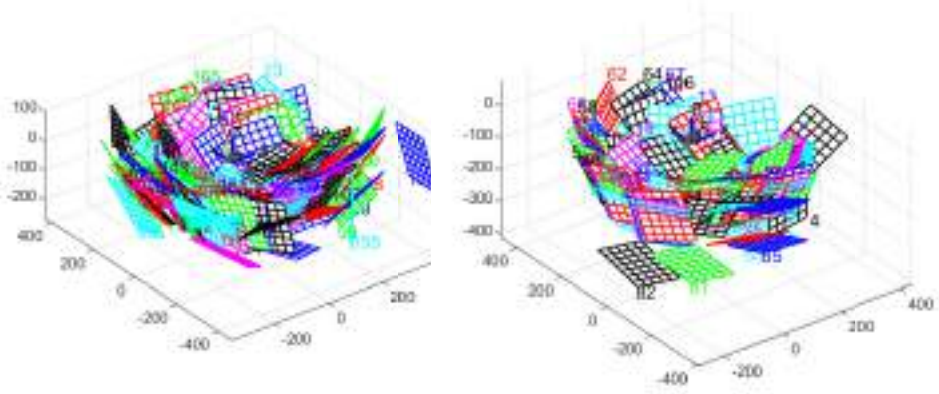


Figura 3.8: Se observa cómo los puntos reproyectados (color magenta) no se encuentran cerca de las esquinas detectadas (color rojo).

Además, en la figura 3.9 se pueden ver las posiciones del patrón de calibración, para cada conjunto/cámara, de las imágenes con las que se estimaron los parámetros intrínsecos con esta herramienta.



(a) Cámara delantera (b) Cámara trasera

Figura 3.9: Parámetros extrínsecos.

3.4.1 Modelo de cámara

En este modelo, se identifican dos planos, uno hipotético que es ortogonal al eje del espejo (u, v) y otro que coincide con el CCD de la cámara (u', v').

La relación entre el vector, $\vec{P_C}$, que parte del punto de vista único ($\boldsymbol{O_C}$) hacia el

(a) (b)

Figura 3.10: Ejemplo de imágenes utilizadas en el proceso de calibración: (a) nueve imágenes *fisheye* del primer conjunto de imágenes (lente delantera) y (b) nueve imágenes *fisheye* del segundo conjunto (lente trasera).

punto 3D de la escena ($\boldsymbol{P}$), y el punto en el plano sensor ($\boldsymbol{m}$) viene dada por:

$$\vec{P}_c = \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} \alpha \cdot u \\ \alpha \cdot v \\ f(u,v) \end{bmatrix} \tag{3.10}$$

donde α significa que los dos ejes del sistema de referencia del plano hipotético son proporcionales a los ejes X_C e Y_C del sistema de referencia 3D con origen en el punto de vista único. Esto se cumple porque se asume que los ejes de la cámara y del espejo se encuentran perfectamente alineados. Para simplificar, incluyen α en la función f de forma que la ecuación (3.10) queda de la siguiente forma:

$$\vec{P}_c = \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(u,v) \end{bmatrix} \tag{3.11}$$

Además de esto, se asume que la función de imagen es rotacionalmente simétrica con respecto al eje del sensor, de modo que la función f dependerá de la distancia del punto al centro del plano, es decir, $f(u,v) = f(\rho)$ siendo $\rho = \sqrt{u^2 + v^2}$. Este modelo se caracteriza por proponer que la función f sea un polinomio cuyo coeficientes y grado se estiman durante el proceso de calibración.

Como se ha comentado, hasta ahora este modelo asume que los ejes de la cámara y el espejo estaban perfectamente alineados. Sin embargo, en realidad puede existir una pequeña desviación. Como consecuencia, los autores proponen utilizar una matriz afín para modelar errores de desalineación y los artefactos de digitalización. Tras aplicar esta transformación afín al punto en el plano hipotético ($\boldsymbol{m} = [u; v]$), se obtiene la proyección en el segundo plano mencionado ($\boldsymbol{m'} = [u'; v']$).

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{3.12}$$

donde c, d y e son los elementos de la matriz afín, y c_x y c_y son las coordenadas del centro de la imagen. Todo esto se puede ver visualmente en la figura 3.11. En la misma página web en la que se encuentra disponible esta herramienta [263], los autores también proporcionan las funciones para proyectar un punto 3D sobre el plano imagen y el proceso inverso. La función que realiza la reproyección de un punto del plano imagen (proceso inverso) devuelve las coordenadas sobre una esfera unitaria ($\vec{p}$) cuyo centro coincide con el punto de vista único ($\boldsymbol{O_C}$).

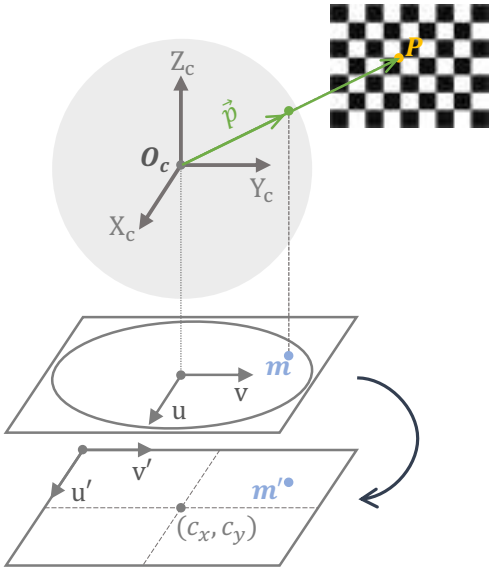


Figura 3.11: Modelo de cámara de Scaramuzza et al. [10].

Tabla 3.9: Parámetros intrínsecos del modelo propuesto por Scaramuzza et al. [10] así como los parámetros relativos al proceso de calibración. Tanto para la cámara delantera (cam0) como la trasera (cam1).

		cam0	cam1
Número de imágenes utilizadas en la calibración		156	112
Tamaño imagen	altura	3000	3000
	anchura	3008	3008
Elementos matriz afín	c	1.0003	0.9999
	d	$1.0219e-04$	$6.3410e-05$
	e	$3.7139e-05$	$-4.2022e-05$
Coordenadas del centro	c_x	1500.6	1492.2
	c_y	1490.6	1492.7
Coeficientes del polinomio	a_0	-666.6782	$-6.671138e+02$
	a_1	0	0
	a_2	$3.3670e-04$	$3.123604e-04$
	a_3	$-4.6828e-08$	$5.795678e-09$
	a_4	$6.7865e-11$	$4.181360e-11$
Error medio de reproyección (en píxeles)		0.909044	0.753854

3.5 Conclusiones

En este capítulo se ha presentado la cámara Garmin VIRB 360, que es la escogida para esta tesis, y además se ha llevado a cabo su calibración mediante dos herramientas *Open Access*: Basalt [38] y OCamCalib [10].

Por un lado, la herramienta de calibración Basalt se ha empleado escogiendo cuatro de los modelos que tiene implementados: (a) modelo de cámara unificado (UCM [16], apartado 3.3.1), (b) modelo de cámara unificado extendido (UCM [4], apartado 3.3.2), (c) modelo de cámara de doble esfera (DSCM [3], apartado 3.3.3) y (d) modelo de cámara Kannala-Brandt (KB [7], apartado 3.3.4). Tras estudiar y comparar los parámetros relativos al proceso de calibración, el modelo KB con ocho parámetros es el que mejor comportamiento ha tenido con respecto al error de reproyección.

Por otro lado, con la herramienta OCamCalib, se realizó el proceso de calibración de forma independiente para cada sistema (sensor CMOS + lente *fisheye*). De modo que, en este caso, se tienen dos errores de reproyección, uno para cada proceso. Si comparamos esto, el error medio de reproyección es menor con la cámara trasera que con la delantera.

Cabe destacar, que tal y como se ha indicado al describir el proceso de adquisición de las imágenes, la calibración con Basalt se ha realizado con fotogramas de un vídeo mientras que con OCamCalib se han empleado imágenes configurando la cámara para ello (modo de fotografía). Teniendo en cuenta que la resolución no es la misma, como trabajo futuro sería interesante emplear la herramienta Basalt con imágenes directas, es decir configurando la Garmin en modo de fotografía en vez de vídeo, tal y como se ha realizado con OCamCalib. Así se podría realizar un estudio comparativo entre todos los modelos de cámaras. Además, también sería interesante como futuro trabajo utilizar dichos modelos en alguna aplicación de robótica para comparar su funcionamiento en base a otra medida que no sea el error de reproyección.

Como se verá en el resto de capítulo, el modelo escogido para esta tesis es el propuesto por Scaramuzza et al. [10]. Esto se debe a que, por un lado, es el modelo que proporciona el *dataset* que se emplea en el capítulo 4. Por otro lado, en el capítulo 5, el *dataset* también se compone de imágenes capturadas configurando la cámara en modo de fotografía. Esto es así porque es la única forma de poder capturar en la misma pose un par de imágenes fisheye (modo RAW) y una vista completa generada de forma interna por la cámara (modo 360).

4

Correspondencia de características en base a métodos probabilísticos

La búsqueda de características locales coincidentes entre imágenes es una técnica crucial en una gran cantidad de aplicaciones de procesamiento de imágenes y visión artificial, como por ejemplo en la localización visual. Un aspecto importante a conseguir con esta técnica es encontrar correspondencias con el menor número posible de falsos positivos. Por dicho motivo, en este capítulo se presentan varias contribuciones en el marco del método *Adaptive Probability-Oriented Feature Matching* (APOFM) [1], con la finalidad de lograr una búsqueda de correspondencias más robusta y orientada a la implementación de un proceso de odometría visual para estimar la trayectoria seguida por el robot. Este método se caracteriza por modelar, de forma dinámica, el entorno, con valores de probabilidad de existencia de correspondencias, es decir, define la repetibilidad y el carácter distintivo de los puntos de la escena. Además, se evalúa su rendimiento cuando las imágenes son tomadas por una cámara con lente *fisheye* y por un sistema de visión catadióptrico, comparando de este modo dos sistemas omnidireccionales. El objetivo principal del presente capítulo es conseguir una búsqueda de correspondencias que sea robusta, implementarla en un algoritmo de odometría visual basado en características locales y realizar un análisis exhaustivo del desempeño de los métodos desarrollados, lo cual llevará a conclusiones relevantes no solo en cuanto a estos métodos, sino también en cuanto a las ventajas e inconvenientes de cada tipo de sistema de visión omnidireccional.

4.1 Introducción

La comunidad científica ha prestado mucha atención en los últimos años a la creación de modelos visuales de entornos, gracias a las numerosas aplicaciones que

existen en una gran variedad de ámbitos, como la robótica móvil [264-267].

En ciertas ocasiones, el robot móvil tiene que actuar en un entorno a priori desconocido [268], en cuyo caso una de las tareas que resultan fundamentales es la creación de mapas. En este sentido, en la actualidad se emplean con gran frecuencia sistemas de visión apoyados en técnicas de procesamiento de imágenes y visión artificial. En particular, el emparejamiento de características [190, 269] otorga la capacidad de encontrar, modelar y rastrear información visual importante del entorno. Una vez realizado lo anterior, el robot móvil se encuentra en disposición de resolver los problemas de mapeo y localización.

Modelar el entorno significa construir una representación del mismo. Como ya se ha descrito en el apartado 2.5, el entorno puede ser modelado principalmente por tres tipos de representaciones: métricas [270, 271], topológicas [113, 272] e híbridas [273]. En la robótica, entre las representaciones métricas, el mapa de ocupación es uno de los formatos más empleados [274]. Esta representación se caracteriza por discretizar el entorno en celdas en las que se codifica uno de los dos posibles estados: ocupado, si hay presencia de algún obstáculo, o libre, en caso contrario.

No obstante, los enfoques clásicos para generar un mapa de rejilla de ocupación tienen ciertas limitaciones, entre las que se encuentran la de no tener en cuenta las correlaciones estructurales entre los diferentes puntos del mapa. Con el objetivo de superarlos, se han aplicado nuevas técnicas como el proceso gaussiano (GP, siglas del inglés *Gaussian Process*) [6], que es un enfoque bayesiano no paramétrico empleado para resolver problemas de regresión y clasificación probabilística.

Este método de aprendizaje es una herramienta potente para, partiendo de datos experimentales, conseguir una identificación precisa de un modelo matemático complejo. La principal ventaja es que combate tanto el ruido del sistema como la incertidumbre del modelo. En relación con esto, O'Callaghan y Ramos [275] presentan un algoritmo denominado *Gaussian Process Occupancy Mapping* (GPOM), en el que, como indica su nombre, emplean esta técnica (GP) para generar una representación de ocupación continua del entorno. Más tarde, Jadidi et al. [276] extendieron el trabajo anterior para crear un mapa semántico, formulando el problema como una clasificación multiclase en vez de como una clasificación binaria (ocupado o libre).

Sin embargo, esta técnica de aprendizaje no tiene como única aplicación la construcción de un modelo del entorno de rejilla, sino que, dentro del campo de la robótica, el proceso gaussiano puede emplearse para resolver varios de los problemas presentes. Por ello, últimamente se ha convertido en una herramienta muy popular entre los investigadores del campo que nos ocupa [277-283]. Por ejemplo, tal y como exponen Nguyen et al. en [284], el proceso gaussiano permite deducir el espesor de la pared en secciones de la tubería que no son vistas por el robot móvil que navega por el interior de esta, teniendo como tarea inspeccionar el lugar de una rotura.

En cuanto a la navegación de un robot móvil, las tareas de creación de mapas y localización pueden desempeñarse siempre y cuando el robot móvil adquiera información de su entorno. Por dicho motivo, el robot móvil se suele equipar con uno o más sensores. Como se ha visto en el apartado 2.2, existen varios tipos (como GPS, *encoders*, sonar o LiDAR). En los últimos años, se ha recurrido con frecuencia, de entre todos los tipos

de sensores que existen, a los sensores de visión. La principal razón está en las diversas características atractivas que presentan, como la riqueza de la información captada o especificaciones físicas como el bajo peso, consumo y tamaño, todo ello con un bajo coste [285]. Además, presentan otra característica relevante para la robótica móvil y es que pueden utilizarse cuando la navegación se realiza en entornos de interior, pero también cuando se produce en un entorno exterior. Aunque la principal ventaja, frente a otros sensores, es la cantidad de información (como textura, formas, color y luminancia) que una sola imagen proporciona del entorno. Como consecuencia, se incrementa el rango de aplicaciones de la robótica móvil en las que se pueden emplear. Aparte de para resolver problemas de localización y creación de mapas, también pueden emplearse, debido al tipo de información que ofrecen, para otras tareas como identificación de obstáculos [286], detección de carreteras [287] y reconocimiento de señales de tráfico [288]. Si bien es cierto que la cantidad de propiedades del entorno que es capaz de adquirir un sistema de visión es una gran ventaja, también resulta de gran interés conseguir un amplio campo de visión del entorno. Esto es posible con un tipo de sistema de visión llamado omnidireccional.

Los sistemas de visión omnidireccionales son capaces de capturar un amplio campo de visión en una única imagen. Esta imagen puede proporcionar una vista de hasta 360 grados del entorno [289]. Como consecuencia, se necesitará un menor número de imágenes capturadas para conseguir un modelo exhaustivo del mismo [290].

Para conseguir un sistema de visón omnidireccional, existen varias configuraciones posibles [125, 291]. En la robótica móvil, las configuraciones más habituales son las conocidas como cámaras dióptricas y catadióptricas. Ambas configuraciones se caracterizan por combinar una cámara convencional con un elemento que amplía el campo de visión, el cual puede ser o bien una lente de gran angular (dióptricas), o bien un espejo (catadióptricas). En cuanto al espejo, este suele ser esférico [292], cónico [293], hiperbólico [294], parabólico o elíptico.

Las cámaras *fisheye* representan una mejor elección que las catadióptricas para algunas aplicaciones (por ejemplo, robots aéreos autónomos), ya que alcanzan una cobertura omnidireccional con un menor peso [295]. La industria del automóvil muestra un interés creciente en las cámaras *fisheye*, con el fin de proporcionar al conductor una visión de 360 grados alrededor del vehículo. Para ello, es habitual hacer uso de cuatro cámaras *fisheye* que deben colocarse de forma que se logre dicha cobertura. En este caso, un sistema catadióptrico capturaría más información irrelevante como cielo y carrocería. Como ejemplo a lo comentado, Lee et al. [296] han posicionado cuatro cámaras *fisheye* (mirando hacia delante, hacia atrás, hacia la izquierda y hacia la derecha) en un vehículo para implementar su algoritmo de cierre de bucle con grafo de pose sin estructura.

La localización global consiste en estimar la pose del robot sin ningún conocimiento previo de esta pose. Los GPS son útiles para esto, pero, se podrían usar otros sistemas como láser o imágenes, que también pueden darnos una estimación global si conocemos el mapa del entorno. Por otro lado, la localización local requiere de una estimación previa de la pose del robot, y va refinando sobre esa estimación previa a partir de los datos capturados por los sensores.

La odometría visual es una de las técnicas más extendidas para resolver el problema de localización empleando un sistema de visión. El concepto de odometría se refiere la estimación, de forma incremental, del movimiento de un agente. Al emplear información visual, calcula la pose relativa entre dos instantes a partir de los cambios que el movimiento produce en las imágenes [28]. Con esto, la odometría visual supera las problemáticas de la odometría de rueda (con *encoders*), como la que aparece cuando el terreno es irregular o se produce un deslizamiento de la rueda. Además, tiene un mayor rango de robots en los que se puede aplicar, pues no se limita únicamente a aquellos que tengan ruedas. Así mismo, la odometría visual, debido al sensor en el que se basa, se puede usar para localizarse tanto en una navegación en un entorno interior como exterior y es una técnica de coste bajo, además de ser relativamente más precisa. Sin embargo, resolver el problema de localización mediante la técnica de odometría visual puede ser todo un reto cuando el entorno no es muy rico en textura o hay presencia de elementos dinámicos, así como cuando las condiciones de iluminación no son las adecuadas.

La odometría visual se puede clasificar en enfoques basados en características locales, basados en apariencia global o híbridos [297]. Valiente et al. [298] realizan una comparación de los dos primeros a partir de imágenes omnidireccionales.

4.1.1 Contribuciones de este capítulo

A este respecto, el presente trabajo continúa la línea de investigación iniciada en [1], en la que se propone utilizar la técnica de coincidencia de características orientadas a la probabilidad adaptativa (APOFM, acrónimo del inglés *Adaptive Probability-Oriented Feature Matching*) para obtener una búsqueda adecuada de correspondencia de características locales en presencia de valores atípicos. Este método se basa en la existencia de correspondencias, obtenidas en iteraciones anteriores, en el espacio 3D mediante una distribución de probabilidad generada por un proceso gaussiano. Esta información se retroalimenta y, una vez detectados los puntos de características locales, se emplea para seleccionar los puntos candidatos para buscar correspondencias.

Existen diversas aplicaciones en las que se requiere la coincidencia de características como, por ejemplo, el seguimiento de objetos [299], la detección [300], el mapeo [301] y la localización [302] en la robótica móvil. En todos ellos, se puede emplear el método mencionado anteriormente (APOFM). A efectos prácticos, este capítulo se centra en el problema relativo a la localización. Los inconvenientes que aparecen en este ámbito se relacionan con la aparición de elementos dinámicos que distorsionan la estimación de la pose y, en consecuencia, se requiere una búsqueda de características robusta. Teniendo en cuenta que el método APOFM ofrece robustez, su empleo se presenta como una opción muy adecuada en un proceso de odometría visual.

Desde esta perspectiva, en este trabajo se presentan varias contribuciones al APOFM con la finalidad de alcanzar una estimación más precisa en relación con la localización. A continuación, se enumeran estas contribuciones:

1. Se propone una búsqueda ponderada y dinámica de correspondencia con la distribución de probabilidad espacial (ver apartado 4.4.5.1). Todo ello, se lleva a cabo con umbrales estáticos y adaptativos.

2. Además, se implementa un clasificador *k-nearest neighbour* empleando diferentes métricas de distancia con la finalidad de mejorar la selección de posibles candidatos que puedan encontrar su correspondencia (ver apartado 4.4.4).
3. Se propone la automatización del recuento de falsos positivos. Para ello se incluye en el algoritmo un detector automático con esta finalidad, que se basa en la distancia entre puntos en el plano imagen y en su proyección 3D (ver apartado 4.4.1.1).
4. Con el objetivo de determinar la eficiencia de las mejoras propuestas, se realiza una comparativa con respecto a otras implementaciones como un método estándar [303], denotado de ahora en adelante como SM (*Standard Method*), empleado *RANdom SAmple Consensus* (RANSAC) [304, 305], así como con el ya mencionado APOFM [1].
5. Además de lo anterior, se realiza un estudio comparativo del desempeño relativo de un sistema de visión catadióptrico y un sistema con lente *fisheye*, para tratar de conocer bajo qué situaciones funciona mejor cada uno de ellos. A estos efectos, hemos utilizado dos conjuntos de datos de imágenes de código abierto que se encuentran disponibles de forma pública [306].
6. Dado que este algoritmo se basa en puntos característicos, hemos estudiado varios detectores y descriptores de puntos característicos (SURF, ORB, FAST y KAZE) sobre imágenes *fisheye*. Se comparan los resultados de una de las variaciones del método APOFM con búsqueda ponderada frente a un método estándar.

4.2 Estimación de pose relativa entre imágenes

Un problema fundamental en la navegación visual consiste en, a partir de un par de imágenes capturadas en instantes y puntos de vista diferentes, estimar la pose relativa (orientación y posición) de la cámara entre ambos instantes de tiempo en las que se capturaron. A esta técnica, que estima el movimiento a partir de cambios en las imágenes, se la conoce como odometría visual [28, 29], tal y como se ha indicado en la sección anterior.

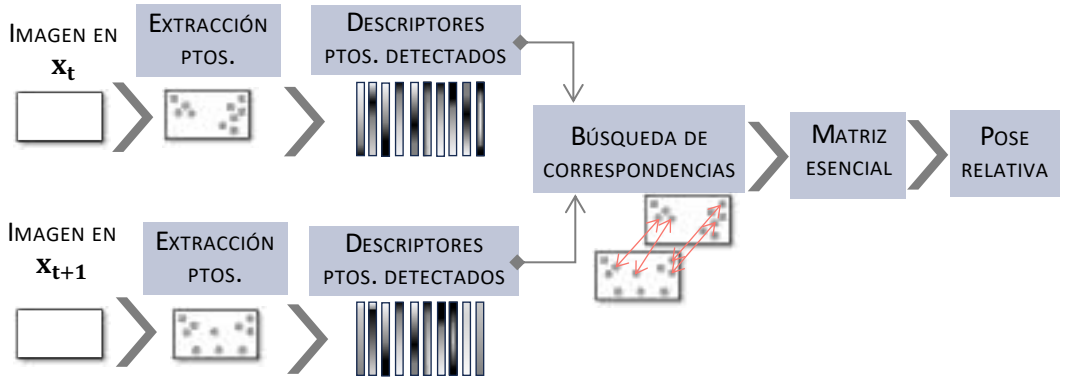


Figura 4.1: Diagrama de bloques de un método estándar para calcular la pose relativa entre un par de imágenes ($\mathbf{I_{t+1}}$ e $\mathbf{I_{t+1}}$).

Como se ha comentado en varias ocasiones en este trabajo, la odometría visual pue-

de resolverse mediante puntos característicos. En este caso, el algoritmo para resolver el problema comentado se compone de los siguientes pasos principalmente, los cuales pueden visualizarse en el diagrama de bloques mostrado en la figura 4.1. Primero, se identifican los puntos de interés en ambas imágenes y se extraen sus correspondientes descriptores. Esta información es la entrada al segundo bloque en el que se realiza una búsqueda de correspondencias entre los dos conjuntos de puntos de interés comparando sus descriptores. El objetivo de este paso es conocer qué puntos 2D representan la proyección del mismo punto 3D y han sido detectados en ambas imágenes. Por último, con esta información, el tercer paso consiste en estimar el movimiento relativo de la cámara mediante dos etapas. Dado que lo que se tiene son correspondencias 2D-2D, el procedimiento consiste en, primero, estimar la matriz esencial y, después, extraer la matriz de rotación y el vector de traslación. A lo largo de este capítulo haremos referencia al algoritmo compuesto por estos pasos como método estándar (SM, siglas del inglés *Standard Method*).

4.2.1 Obtención de pose relativa a partir de la matriz esencial

Cuando se conocen los parámetros intrínsecos de la cámara es posible recuperar la pose relativa mediante la matriz esencial, ya que esta describe la relación geométrica entre dos imágenes. Sin embargo, cabe destacar que, en el caso de la traslación, esta solo puede recuperarse a falta de un factor de escala. Longuet-Higgins introdujo este concepto [307].

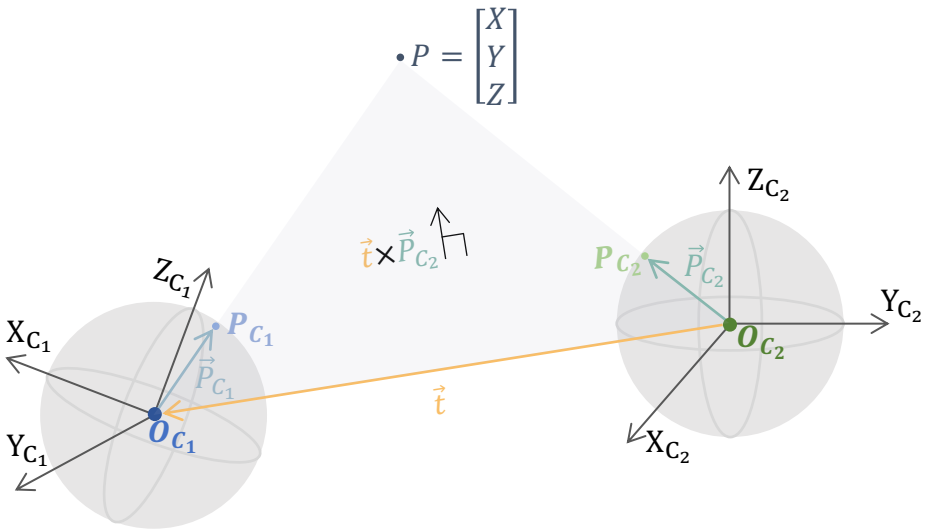


Figura 4.2: Geometría epipolar con modelo esférico para cámaras omnidireccionales, donde $\vec{P}_{C_1}/\vec{P}_{C_2}$ es el vector unitario del rayo que va desde el origen del sistema de referencia de la primera/segunda cámara (centro de la esfera unitaria), $\boldsymbol{O_{C_1}}/\boldsymbol{O_{C_2}}$, al punto 3D ($\boldsymbol{P}$).

Tal y como se muestra en la figura 4.2, si un punto 3D de la escena $\boldsymbol{P}$ es observado desde dos posiciones diferentes ($\boldsymbol{O_{C_1}}$ y $\boldsymbol{O_{C_2}}$), entonces $\vec{P}_{C_1}$ es el vector de dirección (proyección sobre la esfera unitaria) del rayo que va desde el origen del sistema de

referencia de la cámara en la primera posición ($\boldsymbol{O_{C_1}}$) hasta ese punto 3D. Análogamente, $\vec{P}_{C_2}$ es el vector de dirección del rayo cuyo punto inicial es el origen del sistema de referencia de la cámara en la segunda posición ($\boldsymbol{O_{C_2}}$) y su punto final es el punto 3D. De acuerdo con la geometría epipolar, tanto $\vec{P}_{C_1}$ como $\vec{P}_{C_2}$ deben satisfacer la restricción epipolar [303], la cual se define como:

$$\boldsymbol{P_{C_2}}^T \cdot \mathbf{E} \cdot \boldsymbol{P_{C_1}} = 0 \tag{4.1}$$

siendo $\boldsymbol{P_{C_1}}$ y $\boldsymbol{P_{C_2}}$ las coordenadas sobre las esferas unitarias y $\mathbf{E}$ la matriz esencial. Teniendo en cuenta que para calcular la matriz esencial se requiere un conjunto de pares de correspondencias, si este conjunto se define como $\mathbf{A}$, entonces la ecuación anterior puede expresarse como un sistema lineal:

$$\mathbf{A} \cdot \mathbf{e} = \mathbf{0} \tag{4.2}$$

donde $\mathbf{e} = \begin{bmatrix} e_{11} & e_{12} & \dots & e_{32} & e_{33} \end{bmatrix}^T$ son los elementos de la matriz esencial ordenados en un vector columna $\mathbf{e}$. Resolviendo este sistema de ecuaciones homogéneo por descomposición en valores singulares (SVD), se estiman los valores de los elementos de la matriz esencial.

Sabiendo que la matriz esencial únicamente depende de los parámetros del movimiento realizado por la cámara, esta puede definirse como:

$$\mathbf{E} = [\mathbf{t}]_x \mathbf{R} \tag{4.3}$$

siendo $\mathbf{R}$ la matriz de rotación y $[\mathbf{t}]_x$ la matriz asimétrica del vector de traslación $\vec{t} = [t_x, t_y, t_z]$. De este modo, descomponiendo $\mathbf{E}$ en valores singulares, como se describe en [303], se recuperará la pose relativa, mediante la matriz de orientación y la traslación (salvo un factor escala).

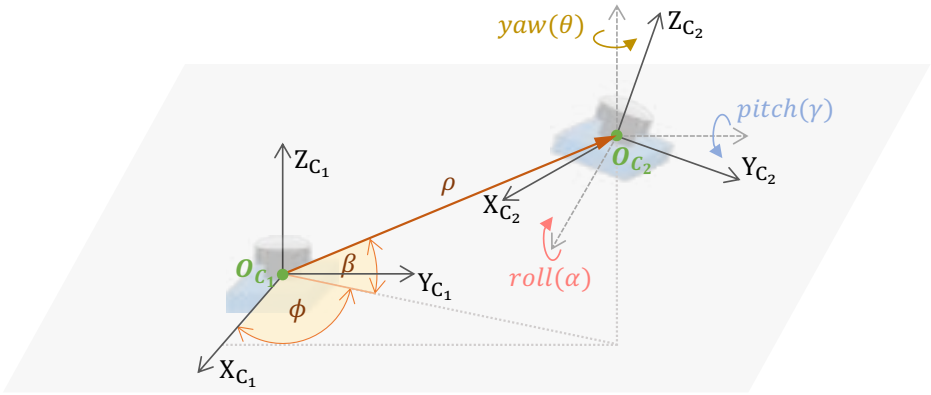


Figura 4.3: Pose relativa entre las dos cámaras donde la traslación viene definida en coordenadas esféricas (β, ϕ, ρ) y la rotación por tres ángulos: θ (*yaw*), γ (*pitch*) y α (*roll*), que representan la rotación alrededor del eje Z, Y y X, respectivamente.

La pose relativa se puede expresar mediante seis parámetros, como se muestra en la figura 4.3, de los cuales cinco son ángulos, tres asociados a la orientación (θ, γ, α) y

dos a la traslación (β, ϕ), y uno de ellos es un factor de escala (ρ) de la traslación. En cuanto a la traslación, las coordenadas cartesianas del vector de traslación (t_x, t_y, t_z) se transforman a coordenadas esféricas (β, ϕ, ρ) mediante las siguientes ecuaciones:

$$\phi = atan2(t_y, t_x) \tag{4.4}$$

$$\beta = atan2(t_z, \sqrt{t_x^2 + t_y^2}) \tag{4.5}$$

$$\rho = \sqrt{t_x^2 + t_y^2 + t_z^2} \tag{4.6}$$

donde ρ es la distancia desde el origen del sistema de coordenadas de la cámara en la primera pose hasta el origen en la segunda pose; β el ángulo de elevación desde el plano X-Y y ϕ el ángulo desde el eje X positivo en el plano X-Y.

Con respecto a la orientación, los tres ángulos se pueden obtener como los ángulos de Euler: *yaw* (rotación alrededor del eje Z), *pitch* (rotación alrededor del eje Y) y *roll* (rotación alrededor del eje X).

4.3 Estimación de pose relativa basada en el modelo de vehículo

El método que se propone en este trabajo se caracteriza por crear un modelo del entorno con información de probabilidad de que las proyecciones de ese punto formen parte de un par de puntos de correspondencia. Para poder emplear esta información, se requiere proyectar desde el sistema del mundo al plano imagen, siendo necesario conocer la transformación del sistema de coordenadas del mundo al de la cámara. Es cierto que este capítulo tiene como objetivo resolver el problema de la odometría visual, por lo que no se conoce la pose de la cámara. No obstante, se puede tener una aproximación inicial de la pose de la cámara para realizar este mapeo de 3D a 2D. Para ello, se asume que la cámara se encuentra a bordo de un robot móvil, así que se puede obtener dicha aproximación mediante el modelo de movimiento de odometría probabilística presentado por Thrun et al. [308]. Atendiendo a esto, los datos de *ground truth* se pueden modelar como datos de odometría (mediante la adición de cierto ruido) y, de este modo, ya se tiene una estimación inicial de la pose en $t + 1$.

El robot móvil se desplaza de t a $t + 1$ y, en esta última posición $t + 1$, la cámara que lleva a bordo captura una imagen $\mathbf{I_{t+1}}$. Así, en ese momento, ya está disponible la información de odometría, proporcionada normalmente por los *encoders* situados en las ruedas. En consecuencia, el modelo de movimiento basado en la odometría puede considerarse una estimación de la pose de la cámara con respecto al sistema de referencia global, pero, en este trabajo, se utiliza únicamente para mapear el modelo 3D y los puntos de la imagen.

4.3.1 Modelo de movimiento de odometría

En el caso de un robot móvil con ruedas, este se mueve por un plano y su estado $\vec{x}$ se representa mediante un punto (x, y) y un ángulo de rotación θ que determina la orientación.

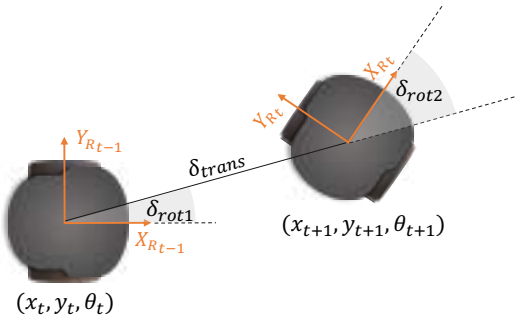


Figura 4.4: El movimiento de un robot basado en odometría se descompone en una rotación (δ_{rot1}), seguida de una traslación (δ_{trans}) y finalmente se produce otra rotación (δ_{rot2}).

El movimiento que ha realizado un robot móvil entre dos poses consecutivas (desde $\boldsymbol{x_t} = (x_t, y_t, \theta_t)$ a $\boldsymbol{x_{t+1}} = (x_{t+1}, y_{t+1}, \theta_{t+1})$ es descrito por el modelo de movimiento basado en la odometría como una secuencia de tres movimientos: una rotación inicial δ_{rot1}, seguida de una traslación en línea recta δ_{trans} y una rotación final δ_{rot2}. Esta secuencia se puede visualizar en la figura 4.4.

Los parámetros del modelo de odometría se pueden calcular a partir de los datos que proporciona el *ground truth* ($\boldsymbol{x_t}$ y $\boldsymbol{x_{t+1}}$) mediante las siguientes ecuaciones:

$$\delta_{rot1} = atan2(y_{t+1} - y_t, x_{t+1} - x_t) - \theta_t \tag{4.7}$$

$$\delta_{trans} = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \tag{4.8}$$

$$\delta_{rot2} = \theta_{t+1} - \theta_t - \delta_{rot1} \tag{4.9}$$

En un caso ideal, los valores del odómetro y los del *ground truth* son iguales. Sin embargo, este hecho no se suele cumplir en condiciones reales, sino que las mediciones proporcionadas por el odómetro corresponden al movimiento real pero cada uno de los parámetros de este movimiento presenta un ruido propio.

Para simular el ruido se utiliza una distribución gaussiana con media cero y varianza σ, que se denomina $\epsilon(\sigma)$. De esta forma los parámetros medidos corresponden a:

$$\hat{\delta}_{rot1} = \delta_{rot1} + \epsilon(\alpha_1 \delta_{rot1} + \alpha_2 \delta_{tras}) \tag{4.10}$$

$$\hat{\delta}_{tras} = \delta_{tras} + \epsilon(\alpha_3 \delta_{trans} + \alpha_4(\delta_{rot1} + \delta_{rot2})) \tag{4.11}$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \epsilon(\alpha_1 \delta_{rot2} + \alpha_2 \delta_{tras}) \tag{4.12}$$

donde α_1, α_2, α_3 y α_4 corresponden a los parámetros que modelan el ruido causado por derivas y deslizamientos (traslación y rotación). Así, el estado del robot $\boldsymbol{x_{t+1}} = (x_{t+1}, y_{t+1}, \theta_{t+1})$ se calcula como:

$$\hat{x}_{t+1} = x_t + \hat{\delta}_{trans} \cos(\theta_t + \hat{\delta}_{rot1}) \tag{4.13}$$

$$\hat{y}_{t+1} = y_t + \hat{\delta}_{trans} \sin(\theta_t + \hat{\delta}_{rot1}) \tag{4.14}$$

$$\hat{\theta}_{t+1} = \theta_t + \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \tag{4.15}$$

La odometría es una técnica de posicionamiento relativo [30]. Sin embargo, para el método propuesto, se necesita una transformación global. La solución a este problema radica en establecer el sistema de referencia global en el estado inicial del robot móvil $\boldsymbol{x}_0$. Así, en el instante $t+1$, la pose del robot móvil con respecto al sistema de referencia global $\mathbf{T_{WR_{t+1}}}$ viene dada por una matriz de rotación alrededor del eje Z, $\mathbf{R_z}(\hat{\theta}_{t+1})$, y una traslación en el plano X-Y, $\vec{t} = (\hat{x}_{t+1}, \hat{y}_{t+1}, 0)$:

$$\mathbf{T_{WR_{t+1}}} = \begin{bmatrix} \cos\hat{\theta}_{t+1} & -\sin\hat{\theta}_{t+1} & 0 & \hat{x}_{t+1} \\ \sin\hat{\theta}_{t+1} & \cos\hat{\theta}_{t+1} & 0 & \hat{y}_{t+1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.16}$$

Si suponemos que la posición de la cámara con respecto al robot móvil $\mathbf{T_{RC}}$ es fija y conocida, entonces la transformación de la cámara en $t+1$ con respecto al sistema de referencia global, ($\mathbf{T_{WC_{t+1}}}$), se puede calcular:

$$\mathbf{T_{WC_{t+1}}} = \mathbf{T_{WR_{t+1}}} \cdot \mathbf{T_{RC}} \tag{4.17}$$

Esta matriz $\mathbf{T_{WC_{t+1}}}$ se empleará para transformar los puntos en el sistema de coordenadas de cámara en $t+1$ al sistema de referencia global. En la figura 4.5, se muestran todos estos sistemas de referencia y sus transformaciones.

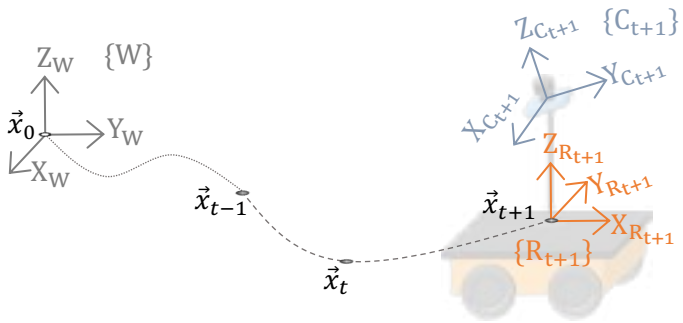


Figura 4.5: Representación de los tres sistemas de coordenadas: $\{\mathrm{W}\}$ (sistema de referencia global), $\{\mathrm{R_{t+1}}\}$ (sistema del robot en $t+1$) y $\{\mathrm{C_{t+1}}\}$ (sistema de la cámara en $t+1$).

4.4 Método APOFM

En este apartado, primero vamos a describir en qué consiste el método APOFM propuesto por Valiente et al.[1], así como las contribuciones de este trabajo para conseguir un comportamiento más robusto.

Dadas dos imágenes de entrada, el objetivo es obtener un conjunto de pares de puntos de correspondencia entre ellas. Durante la búsqueda de correspondencias, se realiza una comparación entre los descriptores de ambas imágenes basada en una medida de similitud. Si un descriptor en una de las imágenes es similar a otro descriptor en otra imagen, entonces los puntos asociados a dichos descriptores son puntos de correspondencia, lo que significa que son la proyección del mismo punto 3D del entorno.

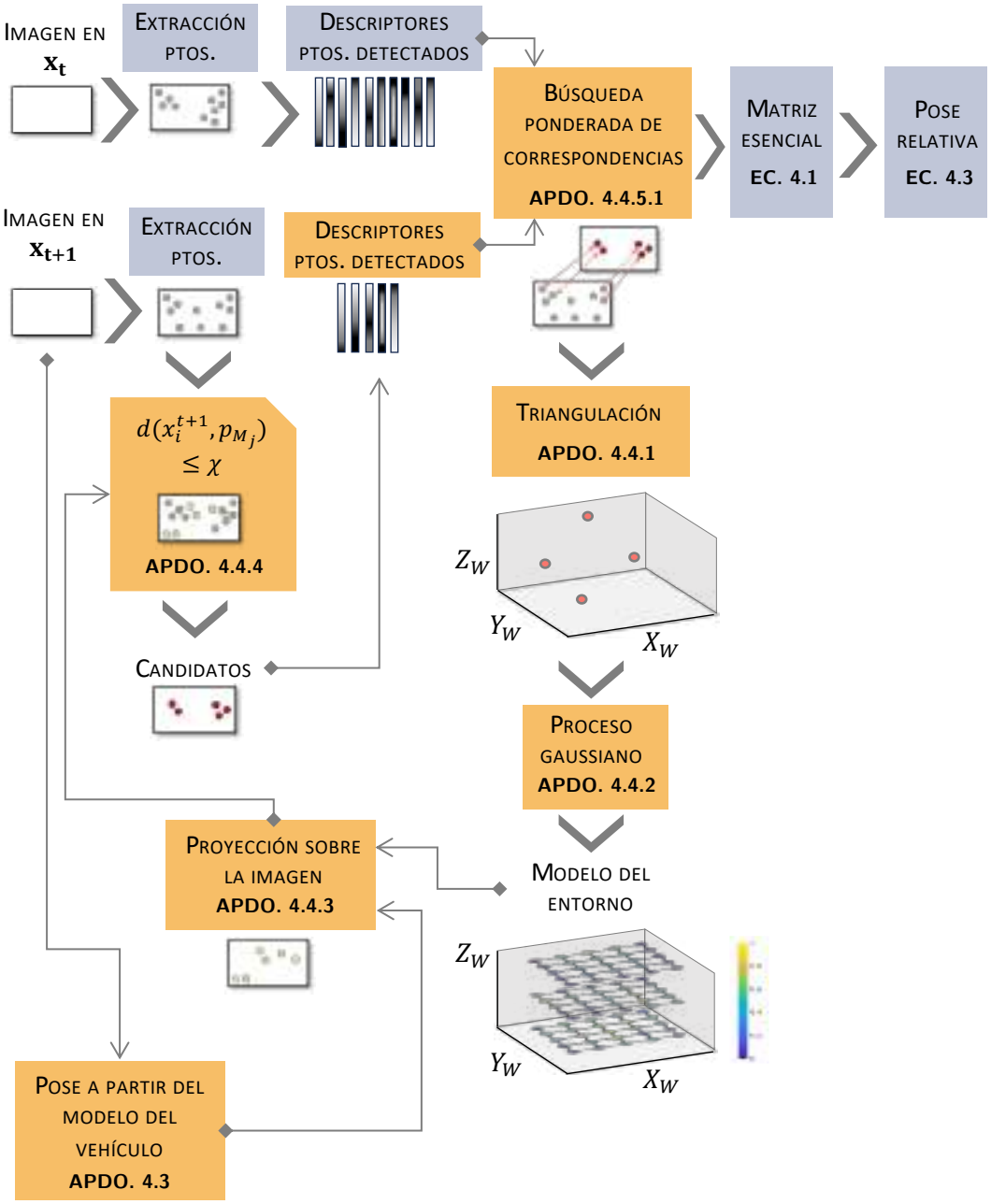


Figura 4.6: Algoritmo para calcular la pose relativa entre un par de imágenes ($\mathbf{I_{t+1}}$ y $\mathbf{I_{t+1}}$) utilizando el método APOFM con las contribuciones de esta tesis. En el diagrama, se encuentran los principales pasos y se indican los apartados donde se explican con más detalle, además de aparecer en otro color (■) para diferenciarlos de los del método estándar (■) (ver figura 4.1).

En la odometría visual, esto se realiza entre la imagen actual y la capturada en el instante anterior para, finalmente, obtener la pose relativa. De este modo, la búsqueda de características se realiza en cada iteración y de forma independiente. Sin embargo, teniendo en cuenta que el algoritmo de odometría visual se aplica mientras se sigue una trayectoria, el par de imágenes que se va a procesar en una iteración específica comparte la información del entorno con la de iteraciones anteriores. Por ende, el méto-

do APOFM, en líneas generales, utiliza esta información común para incrementar la eficacia durante la búsqueda de correspondencias, por lo que existe una realimentación de información en cada iteración.

Esta información se basa en que, si la proyección de un punto 3D ha sido detectada y ha encontrado correspondencia en otras imágenes, se asume que este será robusto ante cambios de puntos de vista y además presentará información visual distintiva del entorno. Siguiendo con esto, ese punto tendrá una alta probabilidad de que también sea detectado en otras imágenes y por ello forme parte de un par de puntos de correspondencia. Por esta razón, el método APOFM se apoya en un modelo 3D del entorno en el que cada punto presenta una probabilidad. Además, esta probabilidad se actualiza en cada iteración del algoritmo de odometría.

El modelo del entorno se obtiene utilizando la técnica de aprendizaje conocida como proceso gaussiano [6] la cual se caracteriza por ser una recopilación de variables aleatorias, de las que, un número finito tienen una distribución Gaussiana conjunta. Cabe mencionar que la entrada es un conjunto de puntos 3D, así pues, el método APOFM dispone de un paso en el que, dado un conjunto de pares de correspondencias, se recuperan las coordenadas de los puntos correspondientes a cada par en el espacio 3D (ver apartado 4.4.1), proceso conocido como triangulación [309]. Tras obtener el conjunto de puntos 3D, este será la entrada al proceso gaussiano que se encargará de actualizar el modelo del entorno con esta nueva información para la siguiente iteración.

Al implementar el método APOFM en un algoritmo de odometría visual, se deben realizar algunos cambios. Por ejemplo, cuando se quiere obtener la pose relativa en el instante $t+1$, la entrada a la etapa de búsqueda de correspondencias no serán todos los puntos característicos detectados y descritos de ambas imágenes ($\mathbf{I_{t+1}}$ y $\mathbf{I_{t+1}}$), sino que, de la imagen actual $\mathbf{I_{t+1}}$, solo se tendrán en cuenta, para buscar correspondencias, aquellos puntos característicos SURF [14] que han sido establecidos como candidatos tras hacer uso del modelo del entorno de APOFM.

Para la selección de candidatos, como se explicará con mayor detalle en el apartado 4.4.4, el modelo 3D obtenido como salida del proceso gaussiano en la iteración anterior se expresa en el sistema de referencia de la cámara del instante de tiempo actual $t+1$ mediante la transformación entre el mundo y el sistema de referencia de la cámara, calculado con el modelo de odometría (ver apartado 4.3). Después, estos puntos se proyectan sobre la imagen $\mathbf{I_{t+1}}$ mediante el modelo de Scaramuzza [10, 310]. En este momento, se tiene un conjunto de puntos 2D proyectados sobre la imagen $\mathbf{I_{t+1}}$ (p) y otro conjunto de puntos SURF detectados en dicha imagen $\mathbf{I_{t+1}}$ $(\mathbf{q})$. Los puntos del segundo conjunto $(\mathbf{q})$ serán clasificados como candidatos en función de su proximidad a los puntos del primer conjunto $(\mathbf{p})$. En la figura 4.6, se muestra, mediante un diagrama, los principales pasos de un algoritmo de odometría visual en el que se ha implementado el método APOFM con las mejoras propuestas en este trabajo.

Como ya se ha comentado, el método APOFM utiliza una realimentación de información conseguida en iteraciones anteriores. Por dicho motivo, en la primera iteración esta información no está disponible, todos los puntos tienen la misma probabilidad de encontrar su punto coincidente. Por consiguiente, en esta primera iteración, se calcula la pose relativa mediante un algoritmo de odometría estándar (i.e. el descrito en

la figura 4.1). En esta iteración se calculará el modelo del entorno tras triangular los pares de correspondencias con los que se ha calculado la matriz esencial. Luego, en la segunda iteración ya se tendrá disponible el modelo del entorno y por tanto se podrá aplicar la búsqueda de correspondencias propuesta en este trabajo.

4.4.1 Triangulación y registro de falsos positivos

Resolver el problema de triangulación implica hallar la posición de un punto en el espacio 3D, del cual se conoce su proyección en al menos dos imágenes. Para poder realizar esto, también deben conocerse los parámetros de calibración y las poses.

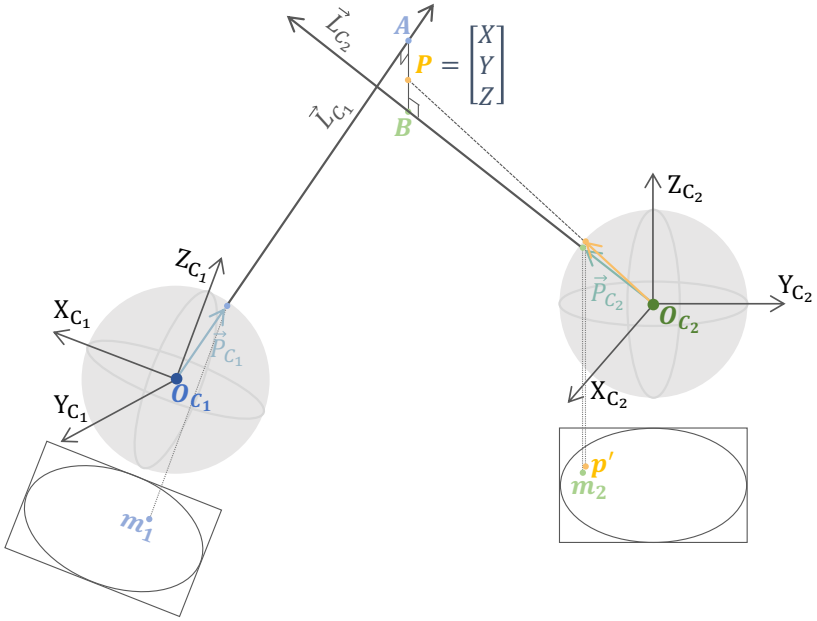


Figura 4.7: Método del punto medio para resolver el problema de triangulación.

El objetivo durante la triangulación es encontrar la intersección de los rayos definidos por sus orígenes, que son los centros de la cámaras ($\boldsymbol{O_{C_1}}$ y $\boldsymbol{O_{C_2}}$), y sus vectores de dirección ($\vec{P_{C_1}}$ y $\vec{P_{C_2}}$), los cuales vienen dados por sus proyecciones sobre la esfera unidad. Todos estos rayos deben estar definidos en el sistema de referencia global para poder obtener la intersección. Ahora bien, esto puede ser una solución no trivial, pues los rayos pueden no coincidir en un punto en el espacio tridimensional por la presencia de ruido en las correspondencias. El origen de este ruido puede ser la distorsión originada por la lente o pequeños errores en la calibración. Asimismo, puede que aparezca ruido derivado del procesamiento de la imagen, ya sea durante la detección de los puntos 2D o por una incorrecta asociación de los mismos. Para resolver el problema de triangulación de forma óptima, hay varios procedimientos propuestos en la literatura, en [311] podemos encontrar algunos. Entre ellos, en este trabajo se ha implementado el método del punto medio, el cual fue propuesto en [312] y establece que el punto 3D es el punto medio del segmento con menor distancia que es perpendicular a ambos rayos. Esto se puede ver en la figura 4.7, donde el punto 3D ($\boldsymbol{P}$) es el punto medio del segmento $\vec{AB}$ definido por los puntos $\boldsymbol{A}$ y $\boldsymbol{B}$, que representan la intersección con los

rayos $\vec{L}_{C_1}$ y $\vec{L}_{C_2}$, respectivamente. Los rayos vienen definidos por su punto inicial $\boldsymbol{O_{C_i}}$ y el vector dirección $\vec{P}_i$ obtenido al proyectar el punto $\boldsymbol{m_i}$ del par de correspondencias sobre la esfera unidad. De este modo, la ecuación correspondiente al primer rayo es $\vec{L}_{C_1} = \boldsymbol{O_{C_1}} + \lambda_1 \cdot \vec{P}_{C_1}$, de igual forma, la del segundo rayo es $\vec{L}_{C_2} = \boldsymbol{O_{C_2}} + \lambda_2 \cdot \vec{P}_{C_2}$.

Como ya se ha comentado en el párrafo anterior, el segmento $\vec{AB}$ es perpendicular a ambos rayos, y, por consiguiente, el producto escalar de su vector director ($\vec{AB} = \boldsymbol{B} - \boldsymbol{A}$) y el asociado a cada rayo será igual a cero.

$$\vec{AB} \cdot \vec{P}_{c_1} = (\boldsymbol{O_{C_2}} - \boldsymbol{O_{C_1}}) \cdot \vec{P}_{c_1} + \lambda_2 \cdot \vec{P}_{c_2} \cdot \vec{P}_{c_1} - \lambda_1 \cdot \vec{P}_{c_1} \cdot \vec{P}_{c_1} = 0 \tag{4.18}$$

$$\vec{AB} \cdot \vec{P}_{c_2} = (\boldsymbol{O_{C_2}} - \boldsymbol{O_{C_1}}) \cdot \vec{P}_{c_2} + \lambda_2 \cdot \vec{P}_{c_2} \cdot \vec{P}_{c_2} - \lambda_1 \cdot \vec{P}_{c_1} \cdot \vec{P}_{c_2} = 0 \tag{4.19}$$

Para poder resolver el problema de triangulación, es necesario hallar los valores de las incógnitas λ_1 y λ_2. Después de esto, el punto 3D requerido ($\mathbf{p}$) se obtiene como el promedio de los puntos $\mathbf{A}$ y $\mathbf{B}$.

$$\boldsymbol{p} = \frac{\boldsymbol{A} + \boldsymbol{B}}{2} = \frac{(\boldsymbol{O_{C_1}} + \lambda_1 \cdot \vec{P}_{c_1}) + (\boldsymbol{O_{C_2}} + \lambda_2 \cdot \vec{P}_{c_2})}{2} \tag{4.20}$$

4.4.1.1 Registro de falsos positivos

En ocasiones, entre los pares de correspondencias puede haber asociaciones incorrectas, es decir, dos puntos se han establecido como proyecciones de un mismo punto 3D cuando realmente no es así, ya que son las proyecciones de dos puntos 3D distintos. A esto se le denomina falso positivo, pues se han considerado como coincidentes cuando en realidad no lo son. Esto puede ocurrir si dichos puntos tienen información visual muy parecida porque, durante la búsqueda de correspondencias, se comparan los descriptores visuales para determinar que esos puntos son coincidentes.

La cantidad falsos positivos es una característica que determina la robustez de la búsqueda de correspondencias. De este modo, con el fin de poder analizar la efectividad del método propuesto, se ha implementado un contador de falsos positivos. Antes de describir en qué consiste, es importante señalar que esta contribución tiene como único objetivo proporcionar información de cuántas correspondencias de las que se están teniendo en cuenta son falsos positivos.

El procedimiento para poder encontrar falsos positivos y contabilizarlos se compone de los siguientes pasos. Dado un punto característico $\mathbf{m_1}$ detectado en la primera imagen y otro punto característico $\mathbf{m_2}$ detectado en la segunda imagen que durante la búsqueda de correspondencias se establecen como coincidentes, se estiman las coordenadas 3D del punto en la escena mediante el procedimiento expuesto en este apartado. Después de ello, este punto 3D es proyectado sobre la segunda imagen ($\mathbf{p}'$). Tal y como se muestra en la figura 4.8, si este punto proyectado ($\mathbf{p}'$) no se encuentra cerca del punto característico $\mathbf{m_2}$, esto significa que $\mathbf{m_1}$ y $\mathbf{m_2}$ no son la proyección del mismo punto 3D, es decir, es un falso positivo.

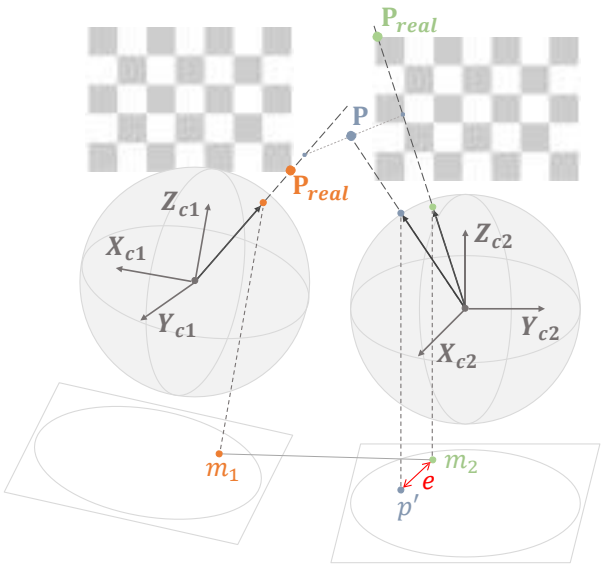
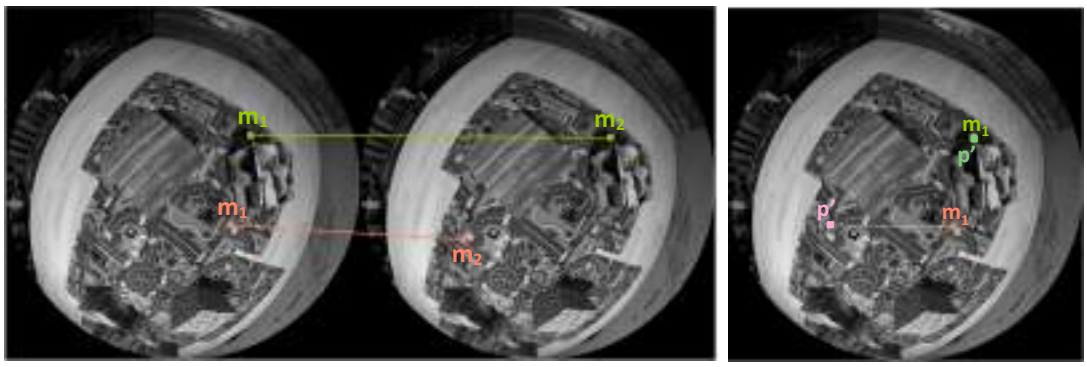


Figura 4.8: Dado un par de puntos de correspondencia compuesto por el punto detectado en la primera imagen (●) y el detectado en la segunda ●, se observa que al proyectar en la segunda imagen el punto 3D obtenido mediante triangulación, este no coincide con el punto detectado en dicha imagen (●).

Teniendo en cuenta la posible presencia de los ruidos comentados al inicio de este epígrafe, para poder determinar si son falsos positivos se calcula la distancia entre $\mathbf{p}'$ y $\mathbf{m_2}$. En el caso de que dicha distancia sea mayor a un umbral establecido, significará que no están cerca y, por consiguiente, se trata de un falso positivo.



(a) (b)

Figura 4.9: En (a) se pueden ver dos pares de correspondencias, cada una de ellas representada con un color e identificada como $\mathbf{m_i}$, donde el subíndice i indica la imagen a la cual pertenece. Después, en (b) se muestran las proyecciones sobre la primera imagen de los puntos 3D recuperados con cada par, donde $\mathbf{p}'$ corresponde al par de correspondencias $\mathbf{m_1 \longleftrightarrow m_2}$, mientras que $\mathbf{p}'$ al par $\mathbf{m_1 \longleftrightarrow m_2}$. Se puede ver que $\mathbf{p}'$ se encuentra cerca del punto característico $\mathbf{m_1}$. Sin embargo, $\mathbf{p}'$ está lejos del punto detectado $\mathbf{m_1}$. Así pues, el par de correspondencias $\mathbf{m_1 \longleftrightarrow m_2}$ es un falso positivo.

En la figura 4.9 se puede visualizar un ejemplo de un par de correspondencias que coincide con la proyección de un mismo punto (verdadero positivo) así como el caso

contrario (falso positivo).

4.4.2 Proceso Gaussiano

En el apartado anterior se ha explicado cómo obtener un conjunto de puntos 3D mediante la triangulación de correspondencias. En este apartado vamos a ver cómo usar esa información para crear una rejilla 3D, que representa el espacio del entorno, muestreado con información de probabilidad obtenida mediante el proceso gaussiano.

El proceso gaussiano (GP, siglas del inglés Gaussian Process) puede entenderse como una variante de la distribución de probabilidad gaussiana para los espacios de funciones. Esto significa que una distribución de probabilidad describe variables aleatorias, mientras que un GP es una distribución sobre funciones. Por tanto, si una distribución gaussiana viene dada por su media y su covarianza, un GP está formado por una función media.

Teniendo en cuenta esto, la ecuación que describe el proceso gaussiano es la siguiente:

$$f(\mathbf{x}) \sim \mathcal{GP}(f_m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{4.21}$$

donde f_m es la función media que depende de un punto de entrenamiento en el espacio dimensional d, $\mathbf{x} \in \mathbb{R}^d$, mientras que k es una función de covarianza que depende de $\mathbf{x}$ y de un punto de test (o consulta), $\mathbf{x}' \in \mathbb{R}^d$.

El modelo probabilístico del entorno se obtiene usando el algoritmo que proponen Jadidi et al. en [313] donde exponen una técnica para el mapeo de ocupación utilizando el proceso gaussiano. El algoritmo se descompone principalmente en tres módulos. En primer lugar, se encuentra la regresión del proceso gaussiano, seguido de un clasificador de regresión logística cuya finalidad es transformar la salida del elemento anterior en valores de probabilidad y, por último, se encuentra una Máquina de Comité Bayesiano (BCM, siglas del inglés *Bayesian Committee Machine*) [314], donde el mapa global se actualiza de forma incremental. Los dos primeros módulos enumerados se describen con más detalle en el apartado 4.4.2.1 y en el apartado 4.4.2.2. En la figura 4.10 se muestra el diagrama de bloques de este algoritmo y en la figura 4.11(a) muestra la salida de este algoritmo, es decir el modelo probabilístico del entorno.

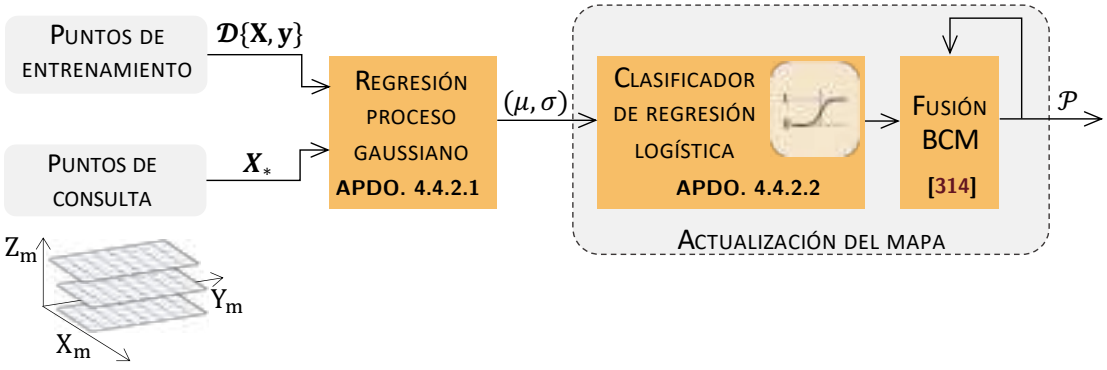


Figura 4.10: Algoritmo basado en GP para calcular el modelo probabilístico del entorno.

4.4.2.1 Regresión de un Proceso Gaussiano

Para la regresión de un proceso gaussiano se tienen dos entradas. Por un lado, se dispone de un conjunto de n puntos de entrada de entrenamiento $\mathbf{X} = \left\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n} | \mathbf{x_i} \in \mathbb{R}^3\right\}$ junto con sus correspondientes valores de salida dispuestos como un vector $\mathbf{y} = \{y_1, y_2, ..., y_n | y_i \in \mathbb{R}\}$ que es la salida de los datos de entrenamiento. Por otro lado, un conjunto de n_t puntos de prueba $\mathbf{X}_* = \left\{\mathbf{x_{*1}}, \mathbf{x_{*2}}, ..., \mathbf{x_{*n_t}} | \mathbf{x_i} \in \mathbb{R}^3\right\}$. La media y la covarianza de la distribución condicional predictiva para los datos de prueba $\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\bar{f}_*, cov(f_*))$ pueden calcularse como sigue:

$$\bar{f}_* = \mathbf{K}(\mathbf{X}, \mathbf{X}_*)^T(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} \tag{4.22}$$

$$cov(f_*) = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}, \mathbf{X}_*)^T(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*) \tag{4.23}$$

siendo σ_n^2 la varianza del ruido de observación y $\mathbf{K}(\cdot, \cdot)$ la matriz de covarianzas de las variables $(\cdot, \cdot)$. Con las covarianzas evaluadas en todos los pares de puntos de entrenamiento $\mathbf{X}$ y de prueba $\mathbf{X}_*$, se obtiene la matriz $\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$ con dimensión $n \times n_t$.

Para el algoritmo que hemos planteado, los datos de entrenamiento $\mathbf{X}$ corresponden a los puntos 3D $(\mathbf{p})$ obtenidos en el paso anterior, es decir, tras resolver el problema de triangulación para cada par de correspondencias de características encontradas. Cada uno de estos puntos de entrenamiento tienen asignado como valor objetivo un valor unitario $(y_i = 1)$. Esta etiqueta significa que dicho punto 3D ha sido detectado en las imágenes en $t-1$ y t y, además, en la búsqueda de correspondencias se han establecido como puntos de correspondencia, formando así un par de correspondencias. Como esto sucede para cada punto 3D de entrada, los datos de salida de entrenamiento y vienen agrupados en un vector de unos $\mathbf{y} = \{1, 1, ..., 1\}$. Por último, los datos de prueba son el conjunto de coordenadas espaciales sobre las que se construye el mapa, que consiste en una malla tridimensional representada por los vectores $\mathbf{x}_m = \{x_1 : i : x_{n_x}\}$, $\mathbf{y}_m = \left\{y_1 : i : y_{n_y}\right\}$, y $\mathbf{z}_m = \{z_1 : i : z_{n_z}\}$ que se definen por un valor inicial y final, y un incremento i entre sus elementos. Este incremento se conoce como el paso de la rejilla $(\Delta grid)$ y si posee un valor bajo producirá un modelo del entorno con mayor resolución (más puntos 3D). Atendiendo a todo lo comentado, el número de puntos de prueba viene dado por la longitud de estos vectores, de modo que $n_t = n_x \cdot n_y \cdot n_z$.

4.4.2.2 Clasificador de regresión logística

La salida del primer elemento de este algoritmo, es decir, de la regresión del proceso gaussiano, es la predicción para los puntos de prueba que viene dada por una media y una covarianza. Teniendo en cuenta que el objetivo es conseguir una representación probabilística del entorno, esta salida debe acotarse para que tenga valores dentro del intervalo $[0, 1]$. Para ello se utiliza una función logística, teniendo:

$$p(y_* = 1|X, y) = \frac{1}{1 + exp(-\gamma\omega_i)} \tag{4.24}$$

siendo $\omega_i = \mu_{*i}\lambda^{1/2}$ la media ponderada, $\lambda = \sigma_{min}^2/\sigma_{*i}^2$ indica la información acotada que se asocia a cada posición, σ_{min} la varianza mínima predicha por la regresión del proceso gausiano, y, por último, γ un parámetro constante positivo que define la forma de la función sigmoide.

4.4.3 Proyección de los puntos del modelo sobre la imagen 2D

Como ya se ha comentado en varias ocasiones durante este capítulo, el método APOFM se basa en el modelo descrito en el apartado anterior para realizar la búsqueda de correspondencias. Por dicho motivo, este modelo debe proyectarse sobre la imagen actual obteniendo así áreas relevantes sobre la imagen. Los píxeles de la imagen que se encuentren en zonas con alta probabilidad encontrarán su correspondencia en la otra imagen también con alta probabilidad. A continuación, se expone el procedimiento para llevarlo a cabo.

Se parte del modelo del entorno obtenido en el apartado anterior, el cual es una distribución 3D de probabilidad $\mathbf{_wP} = \left\{\mathbf{wp_1}, \mathbf{wp_2}, ..., \mathbf{wp_n} | \mathbf{wp_i} \in \mathbb{R}^3\right\}$, cuyos puntos 3D están expresados en el sistema de referencia global. Esto se muestra en la figura 4.11(a). Para poder proyectarlo sobre la imagen, primero hay que aplicar la matriz de transformación para obtener la posición de esos puntos con respecto al sistema de referencia de la cámara.

$$\mathbf{_{C_{t+1}}p_i} = \mathbf{T_{C_{t+1}W}} \cdot \mathbf{wp_i} \tag{4.25}$$

siendo $\mathbf{T_{C_{t+1}W}}$ la matriz que transforma los puntos desde el sistema de coordenadas del mundo al de la cámara en $t+1$. Esta transformación es estimada mediante el modelo del vehículo (apartado 4.3.1). Dado que $\mathbf{T_{C_{t+1}W}}$ es la matriz obtenida con la ecuación (4.17), la ecuación (4.25) se puede expresar como:

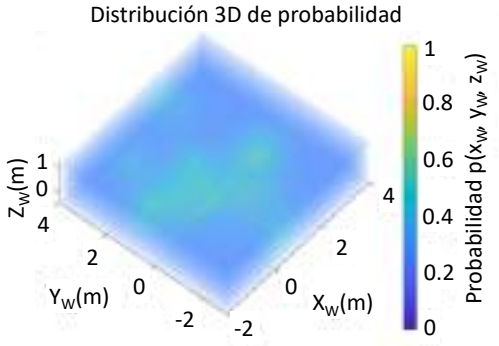
$$\begin{aligned}\mathbf{_{C_{t+1}}p_i} = \mathbf{T}_{C_{t+1}W} \cdot \mathbf{wp_i} = \mathbf{T_{WC_{t+1}}}^{-1} \cdot \mathbf{wp_i} = \\ \left[\mathbf{R_{C_{t+1}W}}^T \quad -\mathbf{R_{C_{t+1}W}}^T \cdot \vec{t}_{C_{t+1}W}\right] \cdot \mathbf{wp_i}\end{aligned} \tag{4.26}$$

siendo $\mathbf{R_{C_{t+1}W}}$ la matriz de rotación que indica la orientación del sistema de la cámara frente al sistema global y $\vec{t}_{C_{t+1}W}$ el vector que une el centro de ambos sistemas de referencia, es decir, la posición relativa expresada en el sistema del mundo. El último paso consiste en proyectar los puntos 3D transformados en el plano imagen mediante el modelo de cámara. El resultado puede verse en la figura 4.11(b), donde los puntos del modelo del entorno aparecen en el plano imagen con su probabilidad asociada. En este ejemplo solo se muestran aquellos con una probabilidad elevada para que se pueda apreciar mejor visualmente.

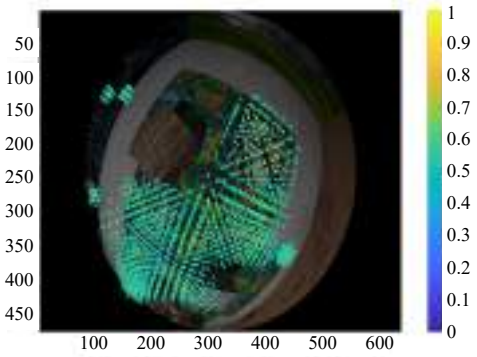
4.4.4 Determinación de puntos característicos candidatos

En este apartado, se va a describir cómo se obtiene el conjunto de puntos característicos que serán la entrada a la búsqueda de correspondencias. Los puntos que forman este conjunto son definidos como posibles candidatos coincidentes usando la distribución de probabilidad proyectada.

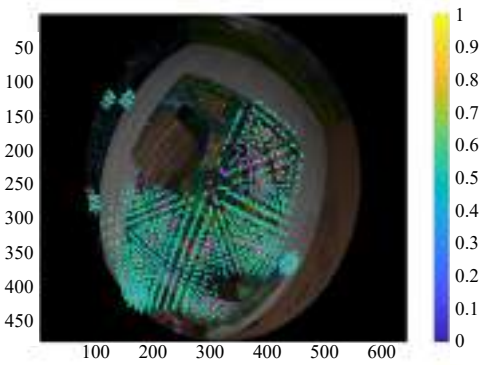
Se consideran que son candidatos aquellos puntos característicos detectados que se encuentran cerca de los puntos de la distribución de probabilidad proyectada. Con este fin, se ha utilizado el método *Nearest Neighbour* [315] ya que se caracteriza por encontrar para cada punto de consulta el punto más cercano en el conjunto de datos de entrenamiento, todo ello en función de la distancia entre los puntos. Además de determinar qué puntos son candidatos, el modelo también se utiliza para asignar a los puntos candidatos la probabilidad asociada al punto más cercano encontrado.



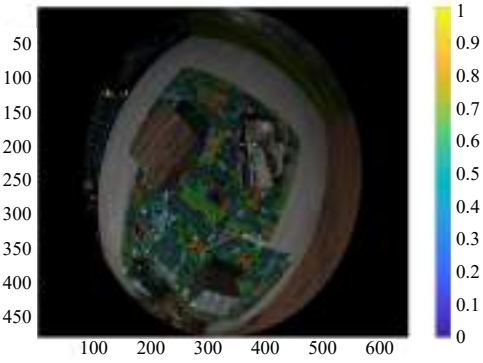
(a) Modelo de probabilidad del entorno que consiste en una distribución de probabilidad 3D de existencia de características (apartado 4.4.2)



(b) Puntos del modelo de probabilidad proyectados en $\mathbf{I_{t+1}}$ (apartado 4.4.3)



(c) Los puntos característicos detectados se representan en la imagen mediante cuadrados en color magenta ■.



(d) Puntos característicos clasificados como candidatos con su probabilidad asociada (apartado 4.4.4).

Figura 4.11: Los puntos de la distribución 3D de probabilidad en el sistema de referencia global (a) son expresados en el sistema de coordenadas de la cámara en el instante $t+1$ y proyectados sobre el plano imagen (b). Una vez detectados los puntos SURF, estos junto con los puntos de la distribución de probabilidad proyectados (c) son empleados para determinar cuántos de los primeros son clasificados como candidatos. Así, finalmente se tiene un conjunto de puntos característicos candidatos junto con su probabilidad asociada.

En este caso, los puntos de entrenamiento en la búsqueda del vecino más cercano son los puntos de la distribución de probabilidad proyectados ($\mathbf{p}$), mientras que el conjunto de puntos de consulta son los puntos característicos detectados ($\mathbf{q}$). La búsqueda del vecino más cercano se realiza de acuerdo con la siguiente ecuación:

$$NN(\mathbf{q}) = \arg\min_{\mathbf{p_i}} d(\mathbf{p_i}, \mathbf{q}) \tag{4.27}$$

Como ya se ha señalado, esta búsqueda se basa en la distancia entre puntos. En la literatura se han utilizado diversas funciones de distancia [316]. Entre ellas se encuentran: euclídea, Mahalanobis, Minkowsky, City-block y Chebyshev.

En este trabajo se han seleccionado dos de estas métricas de distancia (Mahalanobis y City-block) con distintos procedimientos de búsqueda del vecino más próximo. En el caso de la distancia Mahalanobis, se emplea el método exhaustivo para la búsqueda

y por lo tanto se calcula la distancia entre cada punto del conjunto de consulta con los puntos del conjunto de entrenamiento. En el caso de la distancia City-block, la búsqueda se realiza mediante el algoritmo kd-tree [317].

Tras encontrar el vecino más cercano entre ambos conjuntos, el siguiente paso es clasificar cada uno de los puntos característicos detectados (conjunto de consulta) en candidato o no candidato en función de si está cerca o no. Para determinar si está cerca o no, se utiliza la distancia con su vecino más cercano y un umbral que determinará la distancia máxima para clasificar el punto como candidato.

El valor del umbral se obtiene mediante la función de distribución acumulativa inversa chi-cuadrado con n_{dof} grados de libertad y que se evalúa en un valor de probabilidad. En este trabajo, el número de grados de libertad de esta función es igual a 2, que corresponde a la dimensión de los puntos, pues se expresan en el plano imagen. Esta función también depende de otro parámetro de probabilidad cuyo valor será escogido en función de los resultados realizados en el experimento inicial (apartado 4.5.1).

Resumiendo todo lo comentado en este apartado, para un punto característico detectado se buscará si tiene coincidente en la otra imagen si y solo si es clasificado como candidato, o lo que es lo mismo, si cumple la siguiente condición: $d(\mathbf{p_i}, \mathbf{q}) < \chi$. En caso contrario, no serán candidatos y, por lo tanto, se descartan, es decir, no se tienen en cuenta para la búsqueda de correspondencias.

En la figura 4.11(c) se pueden visualizar los puntos característicos detectados (en color magenta) y los puntos de la distribución de probabilidad proyectados. En la figura 4.11(d), se pueden ver qué puntos detectados han sido clasificados como candidatos y qué probabilidad tienen, que se expresa con un color específico dado por un mapa de colores parula, donde el amarillo representa el valor más alto de probabilidad (es decir, uno).

4.4.5 Correspondencias entre imágenes

Dado un par de imágenes, se pueden buscar puntos bidimensionales que son la reproyección de un mismo punto en dos *frames* diferentes. El procedimiento más común para ello consiste en comparar, usando una medida de similitud, todos los descriptores de las características locales detectadas en la primera imagen con todos los de la segunda imagen. Las correspondencias entre conjuntos de características se obtienen encontrando el vecino más cercano en el espacio del descriptor.

La tarea de buscar correspondencias entre un par de imágenes puede plantearse de la siguiente manera [181]. Como entrada se tiene un conjunto de descriptores de características de cada una de las imágenes. Cada uno de los descriptores de características ($\mathbf{f^i_{I_x}}$) tiene una longitud igual a M, y representa un punto característico detectado en una de las imágenes. Por ejemplo, suponiendo que $\mathbf{q^1_{I_1}}$ es un punto detectado en la primera imagen (I_1), su descriptor asociado es $\mathbf{f^1_{I_1}}$. La finalidad es encontrar el punto característico más similar $\mathbf{q^j_{I_2}}$ entre todos los N puntos detectados en la segunda imagen (I_2). Para lograr este objetivo, se compara el descriptor de $\mathbf{q^1_{I_1}}$, $\mathbf{f^1_{I_1}}$, con cada uno de los N descriptores obtenidos en la segunda imagen ($\mathbf{f^j_{I_2}}$). Esta comparación se lleva a cabo con una función de distancia, concretamente con la euclídea.

$$d_j(\mathbf{f_{I_1}^1}, \mathbf{f_{I_2}^j}) = \left\| \mathbf{f_{I_1}^1} - \mathbf{f_{I_2}^j} \right\| = \sqrt{\sum_{i=1}^{M} (\mathbf{f_{I_2}^j}(i) - \mathbf{f_{I_1}^1}(i))^2} \tag{4.28}$$

donde $j = 1, 2...N$. De todas la distancias calculadas, se escoge la distancia mínima obtenida (d_{1st}), siendo el descriptor, con el que se ha obtenido esa distancia, el vecino más cercano ($\mathbf{f_{I_2}^{1st}}$). De este modo, el punto característico asociado a este descriptor será coincidente con $q_{I_1}^1$ solo si esta distancia mínima es menor que un umbral establecido ($u_{matching}$).

Sin embargo, en ocasiones los descriptores no son lo suficientemente discriminatorios y producen correspondencias ambiguas. En esos casos, el descriptor de $\mathbf{q_I^1 1}$ es muy similar a varios descriptores de puntos de la segunda imagen. En consecuencia, la condición anterior no es suficiente. Para resolver esto, se incorpora además una condición basada en *Nearest Neighbor Distance Ratio* (NNDR) [183, 318] además de la condición anterior. Con esta segunda condición, no solo se tiene en cuenta el descriptor más cercano ($\mathbf{f_{I_2}^{1st}}$) sino también el segundo más cercano ($\mathbf{f_{I_2}^{2nd}}$). En este sentido, además de que la distancia más pequeña sea menor que un umbral, la relación entre esta y la segunda distancia menor sebe ser más pequeña que otro umbral. En otras palabras, se debe cumplir el requisito de $d_{1st} \leq u_{matching}$, así como el siguiente:

$$NNDR = \frac{d_{1st}}{d_{2nd}} = \frac{\left\| \mathbf{f_{I_1}} - \mathbf{f_{I_2}^{1st}} \right\|}{\left\| \mathbf{f_{I_1}} - \mathbf{f_{I_2}^{2nd}} \right\|} \leq u_{ratio} \tag{4.29}$$

donde d_{1st} y d_{2nd} son, respectivamente, la menor (vecino más cercano) distancia euclidiana y la segunda menor (segundo más cercano). En el caso de que exista ambigüedad, esta ratio tendrá un valor cercano a uno, debido a que ambas distancias son muy parecidas. Por el contrario, una coincidencia no ambigua tendrá una relación de distancia inferior al umbral [319].

Recopilando todo lo mencionado, el punto de característica local al que corresponda el descriptor más similar, es decir, el que tenga la menor distancia euclídea, formará parte de un par de correspondencias si y solo si: (I) la distancia es inferior a un umbral de coincidencia ($u_{matching}$) y (II) la ratio entre esta distancia y la segunda menor es inferior a un umbral de relación (u_{ratio}).

4.4.5.1 Búsqueda ponderada de correspondencias

Como ya se ha explicado en el apartado 4.4.4, el método APOFM utiliza la distribución 3D de probabilidad para obtener el conjunto de puntos característicos candidatos, que serán aquellos para los que se buscará si tienen puntos característicos coincidentes siguiendo lo descrito anteriormente (apartado 4.4.5). En trabajos anteriores [1] estos candidatos se filtraban, seleccionando únicamente aquellos que tuviesen una probabilidad mayor a una específica.

Sin embargo, en este trabajo proponemos tener en cuenta todos los puntos candidatos y utilizar sus probabilidades asociadas para realizar una búsqueda de correspondencias ponderada y dinámica. Se propone mantener el valor de $u_{matching}$ constante, y que, por el contrario, el valor de u_{ratio} dependa de una función definida. Las funciones

elegidas para ello son tres: escalón, lineal y cuadrática. En la figura 4.12 se pueden ver las tres funciones con las que la búsqueda de correspondencias propuesta será evaluada.

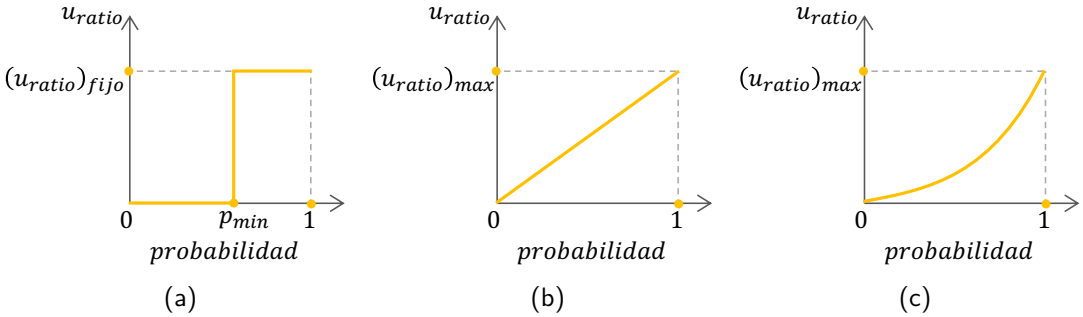


Figura 4.12: Funciones propuestas para la búsqueda ponderada de correspondencias: (a) función escalón definida por un valor mínimo de probabilidad que indica cuándo el umbral u_{ratio} pasa de cero a tomar un valor mayor $(u_{ratio})_{fijo}$; (b) función lineal y (c) función cuadrática. Estas dos últimas tienen un parámetro $(u_{ratio})_{max}$ que indica el máximo valor que el umbral u_{ratio} puede tomar.

En el caso de utilizar, para dicha tarea, la función escalón (figura 4.12(a)), u_{ratio} puede tomar dos valores: (I) cero o (II) un valor fijo superior a cero pero inferior a uno. Tomará un u otro valor en función de si la probabilidad asociada es menor que la probabilidad definida como mínima (p_{min}). Esto significa que aquellos puntos candidatos que presenten una probabilidad menor a p_{min} no se tendrán en cuenta durante la búsqueda coincidente. Por ende, esta función tiene un comportamiento igual que el método APOFM estándar. Por el contrario, cuando se emplea una función lineal (figura 4.12(b)) o cuadrática (figura 4.12(c)) sí se tienen en cuenta todos los puntos candidatos y el valor de u_{ratio} se determina según su probabilidad asociada y la función concreta. Para ambos casos se especifica un valor máximo para u_{ratio}.

4.5 Experimentos y resultados

En este capítulo se han propuesto varias contribuciones con el fin de mejorar el método *Adaptive Probability-Oriented Feature Matching* (APOFM) [1], con el objetivo final de obtener una estimación de la pose más precisa en el marco de una aplicación de odometría visual.

Con la finalidad de poder determinar si efectivamente se ha mejorado el método APOFM estándar, se han realizado una serie de experimentos cuyos resultados se muestran en este apartado. En el primer experimento, se busca encontrar los valores óptimos de dos de los parámetros del método APOFM, que son el paso entre los puntos de la rejilla 3D con la que se define el modelo, Δ_{grid} y el umbral que determina si un punto detectado es candidato, χ.

Con el segundo experimento, se pretende evaluar y comparar los diferentes métodos y variaciones que se muestran en la tabla 4.1. El método APOFM mejorado (descrito en el apartado 4.4.5.1) se evalúa y compara con un método estándar de odometría visual [303] descrito en el apartado 4.2 así como este mismo método estándar empleando RANSAC para eliminar outliers. La comparación con este último se debe a

que la principal característica del método propuesto es su robustez en la búsqueda de correspondencias. El código usado para RANSAC es código abierto que se encuentra disponible en [320] y se ha adaptado para estimar la matriz esencial. Los parámetros de RANSAC han sido seleccionados, tras realizar un análisis, de tal forma que se minimice el error en la estimación de la pose relativa. El estudio de este experimento se ha centrado en los siguientes aspectos: número de características detectadas y de pares de correspondencias (apartado 4.5.2.1), número de falsos positivos detectados (apartado 4.5.2.2) y el error angular (sin información de escala) al estimar la pose relativa (apartado 4.5.2.3).

En el tercer experimento, no se busca evaluar las contribuciones propuestas en este capítulo en sí, sino que está destinado a estudiar otra parte fundamental de este algoritmo, que es la detección y descripción de los puntos característicos. Para ello, se evalúan los mismos aspectos que en el experimento anterior, pero añadiendo el empleo de cuatro métodos distintos para esta tarea. Tanto en los dos experimentos anteriores como en [1], el método escogido es SURF [14], para la detección y para la descripción. En este apartado se sugiere evaluar también otro detector de *blobs*, como es KAZE [41], y dos detectores de esquinas, como son FAST [5] y ORB [11]. Para este experimento, solo se compararán el primer método (SM) y el tercero (WM-SF0.6) de la tabla 4.1.

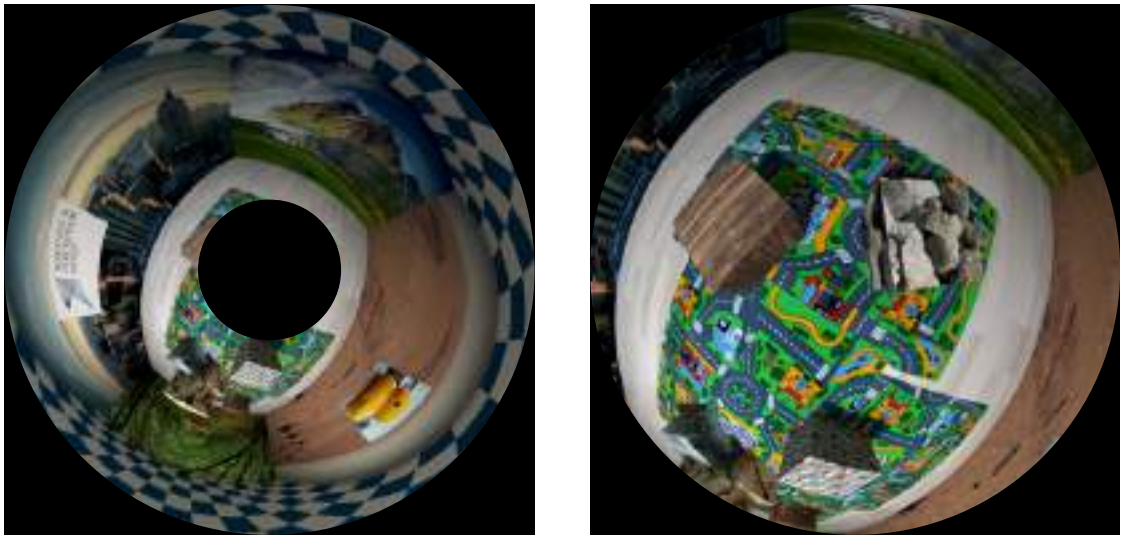
En cuanto a las imágenes, hemos escogido para este capítulo la base de datos de imágenes [321] disponible en [306]. Se compone de imágenes sintéticas generadas con Blender y renderizadas con dos modelos de cámara diferentes (*fisheye* y catadióptrico). La trayectoria seguida por un entorno interior y el número de imágenes es igual para el conjunto de cada uno de los sistemas de visión. Además, todas las imágenes también tienen la misma resolución. Cabe destacar que tanto en el primer (apartado 4.5.1) como en el segundo experimento (apartado 4.5.2), se utilizan ambos conjuntos, pero por el contrario, en el tercer experimento (apartado 4.5.3) solo se utiliza el conjunto de imágenes renderizadas siguiendo la proyección de una cámara *fisheye*. En la figura 4.13 se puede ver un ejemplo de imagen disponible en el *dataset* para cada uno de estos sistemas de visión.

Atendiendo al párrafo anterior, tanto el primer experimento (apartado 4.5.1) como el segundo (apartado 4.5.2) se han llevado a cabo en dos partes ya que, por un lado, se han empleado imágenes capturadas con un sistema catadióptrico y, por otro lado, imágenes *fisheye*. Los resultados correspondientes al primer tipo de imagen (catadióptrica) se visualizarán en las gráficas que se encuentren a la izquierda en cada una de las figuras, mientras que los correspondientes al segundo tipo de imágenes (*fisheye*) se mostrarán en las gráficas situadas a la derecha.

Teniendo en cuenta que una de las principales aportaciones de este capítulo al método APOFM original [1] es la búsqueda de características ponderada, en este apartado se identificará el método propuesto mediante WM (siglas del inglés *Weighted Matching*) seguido del identificador de la función empleada para ello (por ejemplo, para la cuadrática será SqF que corresponde a las siglas de *Square Function*).

Tabla 4.1: Resumen de los métodos y sus variaciones empleados durante los experimentos de este capítulo.

Identificador	Método	Función de WM	Parámetros de la función
SM	Método estándar [303]	–	–
SM+RANSAC	Método estándar con RANSAC para eliminar *outliers* [305]	–	–
WM-SF0.6	APOFM ponderado	Escalón (figura 4.12(a))	$(u_{ratio})_{fijo} = 0.4$ y $p_{min} = 0.6$
WM-SF0.7	APOFM ponderado	Escalón (figura 4.12(a))	$(u_{ratio})_{fijo} = 0.4$ y $p_{min} = 0.7$
WM-LF	APOFM ponderado	Lineal (figura 4.12(b))	$(u_{ratio})_{max} = 0.4$
WM-SqF	APOFM ponderado	Cuadrática (figura 4.12(c))	$(u_{ratio})_{max} = 0.4$



(a) (b)

Figura 4.13: Imágenes de la base de datos [321]: (a) ejemplo de imagen capturada con la cámara catadióptrica y (b) con la cámara *fisheye*.

4.5.1 Experimento 0: Estudio de los parámetros del proceso gaussiano

En este apartado se van a mostrar los resultados obtenidos en el primer experimento, cuyo objetivo es encontrar los valores óptimos para dos de los parámetros del método APOFM como son $\Delta grid$ y χ.

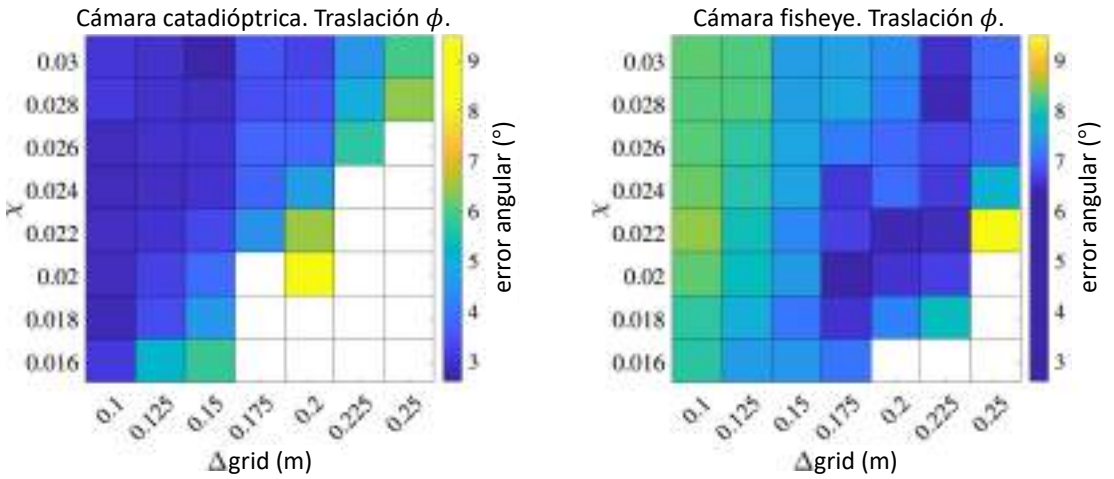
En el primero de ellos, $\Delta grid$, es el paso para construir la rejilla 3D que representa los puntos del entorno (ver apartado 4.4.2.1). Por su parte, el segundo, χ, es el umbral que determina la distancia máxima entre un punto característico detectado y un punto de la distribución de probabilidad proyectada para que el primero sea clasificado como candidato (ver apartado 4.4.4).

Para seleccionar los valores óptimos, en este experimento se evalúa la influencia de ambos parámetros sobre el error al estimar la pose relativa y el tiempo de cálculo. En el caso de la estimación de la pose relativa, nos centramos en el ángulo ϕ de la translación, ya que es el que mayor error tiene asociado. También cabe destacar que únicamente se ha implementado el cuarto método de la tabla 4.1, WM-SF0.7, debido a que es el más restrictivo en cuanto a número de puntos candidatos y, como se analizará posteriormente, si no se consigue un mínimo de puntos de correspondencia, no es posible estimar la pose relativa.

Finalmente, se escogerán los valores de $\Delta grid$ y χ que produzcan, de forma conjunta, un error en el parámetro angular de la traslación y un tiempo de cálculo admisible.

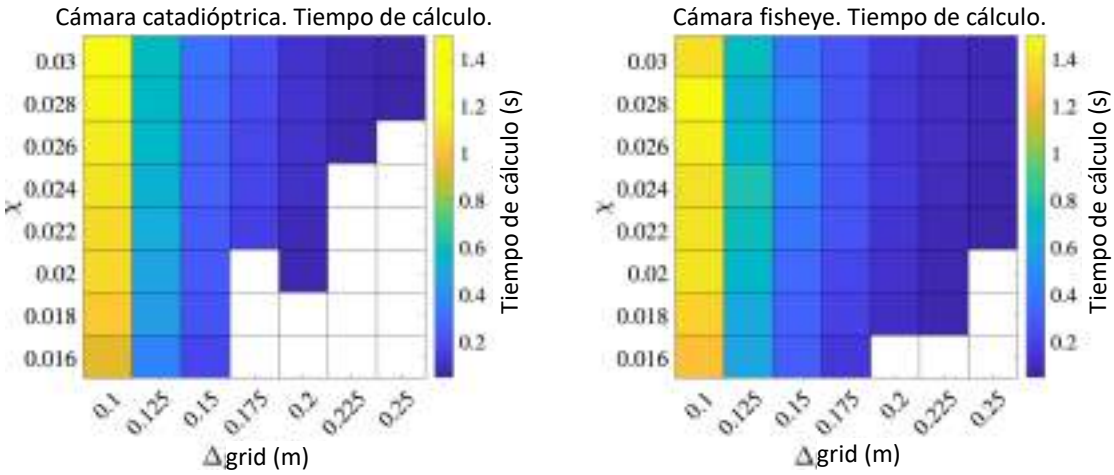
Durante el experimento, se ha implementado el algoritmo con la variación WM-SF0.7 y se ha ejecutado para diferentes valores de $\Delta grid$ y χ, mientras que el resto se mantenían fijos. Cada vez que se ejecutaba el algoritmo con una combinación de valores dados, se calculaba la pose relativa entre cada una de las imágenes del *dataset* (t) frente a sus tres imágenes siguientes ($t+1$, $t+2$ y $t+3$). Para cada combinación, se obtuvo el valor medio del error cometido al estimar la pose relativa. Se observó que

el error cometido al estimar el ángulo acimut (ϕ) de la traslación es mayor que el que se obtiene con el resto de parámetros que describen la pose relativa, de tal forma que se analizarán los resultados de este parámetro.



(a) (b)

Figura 4.14: Influencia de los parámetros χ y $\Delta grid$ en los resultados de estimación del ángulo ϕ de la traslación relativa, cuando se emplea una cámara (a) catadióptrica o (b) *fisheye*



(a) (b)

Figura 4.15: Influencia de los parámetros χ y $\Delta grid$ en el tiempo de ejecución del algoritmo, cuando se emplea una cámara (a) catadióptrica o (b) *fisheye*.

Atendiendo a lo anterior, la figura 4.14 muestra las medias aritméticas de errores angulares cometidos al estimar el ángulo (ϕ) de la traslación para cada combinación. Estos valores vienen representados por un color específico y, en el caso de que no

haya sido posible obtener la matriz esencial por no conseguir el mínimo de pares de correspondencias, este color será el blanco. Se ha observado que esto ocurre cuando se producen dos hechos: por un lado, el valor de χ es pequeño y, por otro lado, el valor de $\Delta grid$ es alto. Esto se traduce en que un valor pequeño de χ conlleva a una selección de candidatos más estricta y, por ende, el número de puntos sobre los que se buscarán correspondencias será menor, y, en consecuencia, más puntos característicos serán eliminados. Además, debido al segundo hecho, el entorno está representado por un número menor de puntos, traduciéndose en una pérdida de información 3D y un menor número de puntos candidatos.

Tras comparar los resultados obtenidos con ambos tipos de sistemas de visón (ver figura 4.14), observamos que el comportamiento es diferente. En el caso de la cámara catadióptrica, figura 4.14(a), cuando el valor de $\Delta grid$ es bajo (el modelo del entorno se compone de más puntos) y χ es alto (mayor número de candidatos), se comprueba que el error en la traslación es bajo. Por el contrario, con la cámara con lente *fisheye*, figura 4.14(b), esto (i.e. menor error) ocurre cuando tanto χ como $\Delta grid$ toman valores que se encuentran en la zona central del intervalo estudiado.

Con respecto al tiempo de cálculo del proceso, los resultados se visualizan en la figura 4.15. Comparando ambos tipos de cámaras, se puede ver que el comportamiento es el mismo, es decir, el tiempo de cálculo es independiente del tipo de imagen, y depende principalmente del parámetro $\Delta grid$. A medida que aumenta este parámetro, el tiempo de cálculo disminuye. Esto era de esperar, ya que a mayor valor de $\Delta grid$, menor es el número de puntos que representan el entorno y, por lo tanto, el proceso gaussiano tiene que procesar menos puntos. Lo mismo ocurre al reproyectar el modelo en el plano imagen. Analizando las gráficas, podemos ver que el tiempo de cálculo también depende del valor de χ aunque la influencia de este es menor que la de $\Delta grid$. En cuanto a la relación, el tiempo de cálculo aumenta cuando lo hace el valor de χ. Esto tiene sentido ya que la selección de puntos candidatos es menos restrictiva, obteniendo más puntos característicos como candidatos y como consecuencia el proceso gaussiano, así como el de la búsqueda de correspondencias, tienen que procesar más puntos.

En resumen, de este experimento podemos concluir que el proceso gaussiano influye más en el tiempo de cálculo que en otras partes de este método como el procesamiento de imágenes.

A partir de todo lo analizado, se han escogido los valores de χ y $\Delta grid$ que proporcionan un buen equilibrio entre error y tiempo de cálculo. Teniendo en cuenta que se usarán los mismos valores para los dos tipos de imágenes, se ha buscado que la relación anterior se cumpla más o menos para los dos tipos de cámaras. Por todo ello, los valores que hemos escogido son, por un lado, $\Delta grid = 0.15$, ya que, para este valor, los resultados presentan un buen equilibrio en cuanto a error y tiempo de cálculo. Aunque es cierto que son mejores cuando se utilizan imágenes capturadas por la cámara catadióptrica, escoger un valor mayor podría conducir a que no se encontrara el mínimo número de correspondencias, no pudiendo calcular la pose relativa. Por otro lado, como se ha comentado, χ no tiene tanta influencia en el tiempo de cálculo, por lo que, una vez seleccionado el valor de $\Delta grid$, nos hemos fijado en la figura 4.14 para seleccionar el valor de χ que suponga el mejor resultado. Esto se produce cuando $\chi = 0.018$. De este modo, el siguiente experimento se realizará con estos valores óptimos.

4.5.2 Experimento 1: Evaluación del método APOFM

Una vez escogidos los valores de $\Delta grid$ y χ óptimos, en este segundo experimento vamos a comparar distintas variaciones del método APOFM con respecto a la aportación relativa a la búsqueda de correspondencias ponderada. Para dicha comparación, se analizarán los resultados con respecto a aspectos de la búsqueda de correspondencias, como es el número de correspondencias (apartado 4.5.2.1) y los falsos positivos (apartado 4.5.2.2), así como de la localización, como es el error al estimar la pose relativa entre pares de imágenes (apartado 4.5.2.3). Además, los resultados de las variaciones del método APOFM se comparan también con los resultados obtenidos con un método estándar y este último utilizando RANSAC.

Este experimento se ha realizado en dos partes. En primer lugar, se han ejecutado los seis métodos (ver tabla 4.1) siendo la entrada al algoritmo imágenes capturadas por un sistema catadióptrico. En segundo lugar, estos mismos métodos se han ejecutado con imágenes *fisheye*. El objetivo de este experimento es evaluar y comparar los seis métodos, pero también los dos sistemas de visión omnidireccionales.

4.5.2.1 Número de pares de correspondencias de características locales

El primer factor a estudiar es el número de puntos característicos SURF detectados en la imagen $\mathbf{I_{t+1}}$ que han sido clasificados como candidatos y cuántos han encontrado correspondencias en la otra imagen $\mathbf{I_{t+1}}$. Cabe destacar que se ha ejecutado tres veces el algoritmo para cada imagen de la base de datos. De esta forma, para cada imagen se han buscado correspondencias y calculado la pose relativa con tres imágenes capturadas en tres instantes de tiempo distintos ($t+1$, $t+2$ y $t+3$) y dado que se sigue una trayectoria, también con distinta distancia (d_1, d_2 y d_3) con respecto a la $\mathbf{I_{t+1}}$. El objetivo es ver la influencia de la distancia entre poses en la búsqueda de correspondencias.

La figura 4.16 muestra el valor medio de estos resultados, donde cada fila corresponde a una distancia y cada columna a un tipo de cámara. La primera fila corresponde a los resultados con la menor distancia, d_1, es decir, los obtenidos tras utilizar como entrada al algoritmo el par de imágenes compuesto por $\mathbf{I_t}$ y $\mathbf{I_{t+1}}$. Por su parte, la segunda fila, d_2, está asociada al par de imágenes de entrada compuesto por $\mathbf{I_t}$ y $\mathbf{I_{t+2}}$. Por último, la tercera fila corresponde a los resultados con la mayor distancia entre imágenes, d_3, es decir, los obtenidos tras utilizar como entrada al algoritmo el par de imágenes compuesto por $\mathbf{I_t}$ y $\mathbf{I_{t+3}}$.

Centrándonos en las gráficas, se ha empleado un diagrama de barras para representar el número de puntos cuyos valores se encuentran definidos en el eje izquierdo de las mismas. La primera barra corresponde al método estándar (identificado como SM). En este caso, como no existe filtrado, la altura de la barra representa los puntos SURF detectados en la imagen I_{t+i}, donde i toma el valor de 1, 2 o 3 en función de la fila en la que se encuentre la gráfica. También se muestra, con una barra superpuesta con menor ancho, el número de puntos detectados que han encontrado otro punto coincidente en la imagen $\mathbf{I_{t+1}}$. La segunda barra representa lo mismo, pero utilizando RANSAC al estimar la matriz esencial (identificado como SM+RANSAC). Las siguientes cuatro barras están relacionadas con el método APOFM y cada una de ellas representa a una variación en cuanto a la función para la búsqueda ponderada de co-

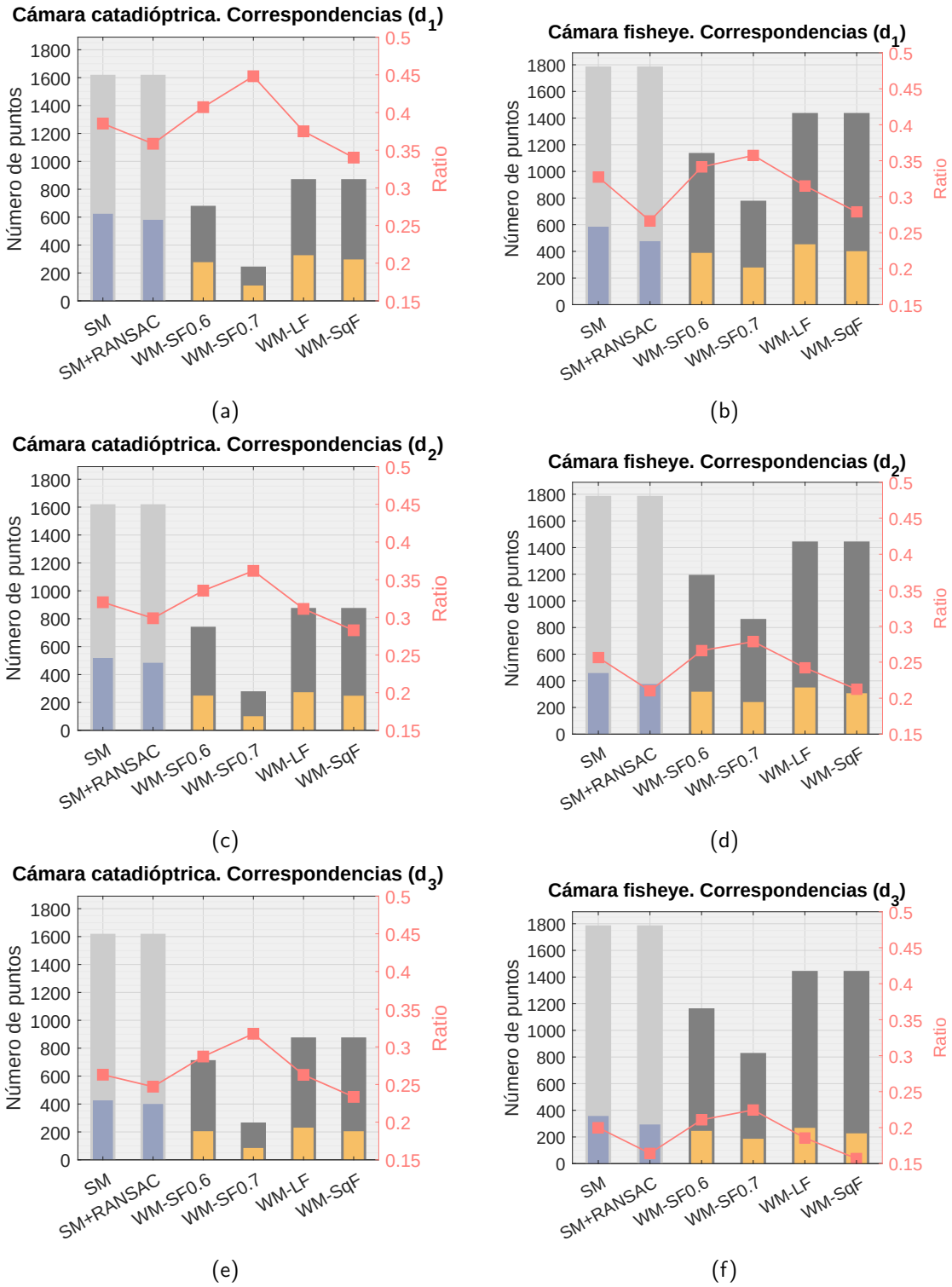


Figura 4.16: El eje izquierdo muestra: el número total de puntos SURF (■); el número de pares de correspondencias con el método estándar (con y sin RANSAC) (■); el número de puntos que han sido clasificados como candidatos con APOFM (■) y cuántos de estos últimos han encontrado correspondencias realizando una búsqueda ponderada (■). En el eje derecho se representa la ratio (—■—) entre los valores representados por las dos barras.

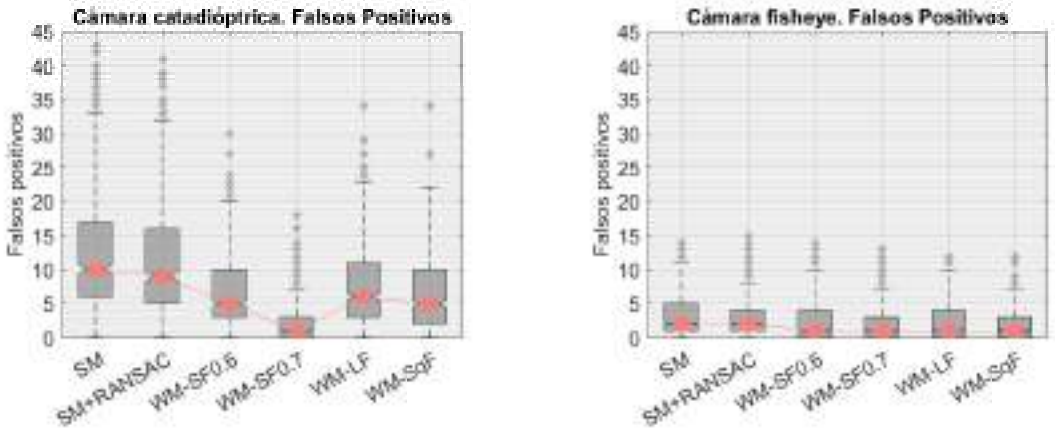
rrespondencias (tabla 4.1): función escalón (SF) con $p_{min} = 0.6$ (WM-SF0.6) y con $p_{min} = 0.7$ (WM-SF0.7); función lineal (WM-LF); y función cuadrática (WM-SqF). De este modo, las alturas de estas barras corresponden a, por un lado, el número de puntos que han sido clasificados como candidatos, y, por otro lado, cuántos de ellos han encontrado correspondencias en la imagen $\mathbf{I_{t+i}}$. Además de todo lo mencionado, en cada gráfica, se representa, mediante una tendencia, la ratio entre el número de puntos de características para los que se ha buscado correspondencia y el número de ellos para los que finalmente se ha encontrado su correspondencia y se han empleado para calcular la matriz esencial. Estos valores corresponden al eje derecho.

Tras analizar los resultados, observamos que, en el caso del método estándar (SM), el número de puntos SURF detectados es mayor en las imágenes *fisheye*. Por el contrario, el número de correspondencias es menor que con las imágenes capturadas con una cámara catadióptrica. Este efecto puede deberse a que el campo de visión es mayor. Siguiendo con el método estándar, pero con RANSAC (SM+RANSAC), en este caso, en número de correspondencias es menor. Esto último era de esperar ya que RANSAC elimina aquellos pares de correspondencias que no se ajusten a la matriz esencial.

En cuanto al método APOFM, se puede ver que se han clasificado más puntos como candidatos en las imágenes *fisheye*. A pesar de esto, la cámara catadióptrica presenta unos valores de ratio más elevados, lo que quiere decir que, aunque hayan más puntos candidatos en las imágenes *fisheye*, muchos de ellos no encuentran sus correspondencias. Este hecho puede deberse a varios factores, como que la selección de candidatos se basa en la distancia en píxel a puntos 3D del entorno que se han proyectado. Siguiendo con esto último, hay que tener en cuenta que al reproyectar un mismo rayo, este puede corresponder a distintos puntos 3D, en función de la profundidad, y, por lo tanto, todos estos puntos son proyectados en la misma posición de píxel. Como consecuencia, puede darse el caso de que al reproyectar un punto 3D de la escena sobre una imagen, el píxel que se encuentra en esa posición no corresponda a dicho punto 3D debido, por ejemplo, a oclusiones. Por dicho motivo, se debe realizar la búsqueda de correspondencias comparando los descriptores locales, resultando en que muchos puntos candidatos no encuentren otro punto en la otra imagen. Siguiendo con la comparación de los descriptores, también podría deberse a que SURF funciona mejor en las imágenes catadióptricas que en las *fisheye*.

En cuanto a cada una de las variaciones del APOFM, los mejores resultados en cuanto al ratio y englobando ambos tipos de imágenes se han obtenido con el tercer método (WM-SF0.7). A pesar de esto, con los resultados mostrados hasta ahora no podemos obtener una conclusión en cuanto a eficacia, pues se debe analizar el número de falsos positivos, así como el error al estimar la pose relativa.

Para terminar con el estudio sobre el número de puntos, como se ha comentado, los resultados se han dividido en función de tres distancias que corresponden a tres instantes de tiempo. Así, discutimos cuál es su influencia. Tras analizar los resultados mostrados en la figura 4.16, se comprueba que a mayor distancia el número de correspondencias es menor y esta influencia es independiente del tipo de cámara, aunque para el caso de la cámara catadióptrica se obtienen un menor número de correspondencias que con la cámara *fisheye*, a una misma distancia. Esto puede venir motivado porque el campo de visión es mayor y, por tanto, la diferencia de información visual entre dos imágenes



(a) (b)

Figura 4.17: Se muestra el número de falsos positivos, detectados para cada par de imágenes, mediante un diagrama de caja. Además, también se representa la media aritmética (—■—) para cada caso del eje X. En (a) se visualizan los resultados obtenidos al emplear imágenes capturadas con la cámara catadióptrica y, por el contrario, en (b) los conseguidos con imágenes *fisheye*.

es menor, aunque la distancia entre sus puntos de captura sea mayor.

4.5.2.2 Falsos positivos

El hecho de que se tenga una gran cantidad de correspondencias es bueno, pues se tiene más información para el cálculo del movimiento relativo, pero siempre y cuando estas sean correctas, ya que en caso contrario pueden producir una incorrecta estimación del modelo. En resumen, se pretende que la búsqueda de correspondencias sea precisa. Para esto último, es necesario estudiar la presencia de falsos positivos, que es lo que haremos en este apartado.

Los resultados de este estudio se exponen en la figura 4.17, la cual presenta un diagrama de caja para cada uno de los métodos que estamos evaluando (ver tabla 4.1). Para cada iteración (es decir, cada par de imágenes), se ha calculado el número total de falsos positivos presentes mediante el registro descrito en apartado 4.4.1.1. Cada uno de estos valores devueltos por el registro se han ido recopilando, de tal forma que al finalizar el experimento se dispone de la cantidad de los falsos positivos para cada par de imágenes. Esta cantidad viene representada por los diagramas de cajas.

Una vez examinadas estas gráficas, podemos señalar que utilizando las distintas variaciones propuestas en este capítulo se logra un rango de número de falsos positivos menor al obtenido con el método estándar. Centrando el estudio en los resultados con el método propuesto en este capítulo, se observa que la distribución más condensada, la cual supone un menor número de falsos positivos detectados, así como una menor dispersión, corresponde a WM-SF0.7. En cuanto a las dos variaciones del método estándar (SM y SM+RANSAC), podemos ver cómo el número de falsos positivos se ha reducido al emplear RANSAC. A pesar de esto, si lo comparamos con los resultados logrados con el método APOFM, este último consigue una disminución de falsos positivos más evidente. Si comparamos los resultados en función del tipo de cámara, podemos con-

cluir que se ha detectado un menor número de falsos positivos cuando las imágenes de entrada son las capturadas por una cámara *fisheye*. De hecho, si observamos los diagramas de cajas para estas imágenes, podemos ver que cuando el método es una variación del APOFM ponderado, el *whisker* inferior no está presente, lo que significa que el menor número de falsos positivos obtenido es cercano o igual al cuartil inferior (Q1) de todos los valores. El número de falsos positivos detectados en gran parte de pares de imágenes se encuentra entre cero y un valor pequeño.

En resumen, la búsqueda de correspondencias es más robusta cuando se emplea el método propuesto en este capítulo. No obstante, dado que el algoritmo completo tiene como tarea calcular la pose relativa entre cada par de imágenes de entrada, también es preciso llevar a cabo un estudio sobre el error cometido en dicha tarea.

4.5.2.3 Error de localización

Las finalidad de este trabajo es estimar la pose relativa con la mayor precisión posible, por dicho motivo se han propuesto una serie de mejoras para el método APOFM [1]. En los apartados anteriores hemos analizado varias características durante la búsqueda de correspondencias y se ha obtenido que, en estos aspectos, con las aportaciones se alcanzan mejores resultados. De modo que en este apartado vamos a analizar el error cometido al estimar la pose relativa implementando el método APOFM con dichas mejoras (WM-SF0.6, WM-SF0.7, WM-LF y WM-SqF) y, además, se comparan con el método estándar (SM y SM+RANSAC).

Como se mencionó en apartado 4.4.4, se han implementado dos tipos de búsqueda del punto de probabilidad proyectado más cercano a los puntos característicos detectados. Las dos posibilidades son (a) una búsqueda exhaustiva con distancia de Mahalanobis y (b) algoritmo de búsqueda kd-tree con distancia City-block. Ambas serán analizadas en este apartado, así pues, en la figura 4.18 y en la figura 4.19, para cada método basado en APOFM se tendrán dos barras, una para cada una de estos enfoques de búsqueda para la selección del conjunto de puntos candidatos. Por el contrario, tanto SM como SM+RANSAC tendrán una única barra, siendo la primera y la segunda respectivamente.

La figura 4.18 muestra el error medio cometido en los dos parámetros angulares correspondientes a la traslación. La primera fila representa los resultados asociados al parámetro angular (ϕ) con una cámara catadióptrica (figura 4.18(a)) y con una *fisheye* (figura 4.18(b)). Mientras que la segunda fila expone los resultados del segundo parámetro angular (β) con ambos sitemas de visión, catadióptrico (figura 4.18(c)) y *fisheye* (figura 4.18(d)).

Una vez analizada la información de la figura 4.18, podemos concluir que, en cuanto a los sistemas de visión, la traslación calculada a partir de imágenes *fisheye* tiene un error menor que las capturadas con un sistema catadióptrico cuando el algoritmo corresponde a cualquiera de las dos variaciones del método estándar (SM y SM+RANSAC). Sin embargo, ocurre lo contrario cuando se ejecutan todas las variaciones del método APOFM ponderado. En cuanto a los métodos, los resultados muestran que la estimación de la traslación relativa es más precisa utilizando RANSAC que sin él (SM). De hecho, en el caso de las imágenes *fisheye*, el valor de error medio con RANSAC es muy

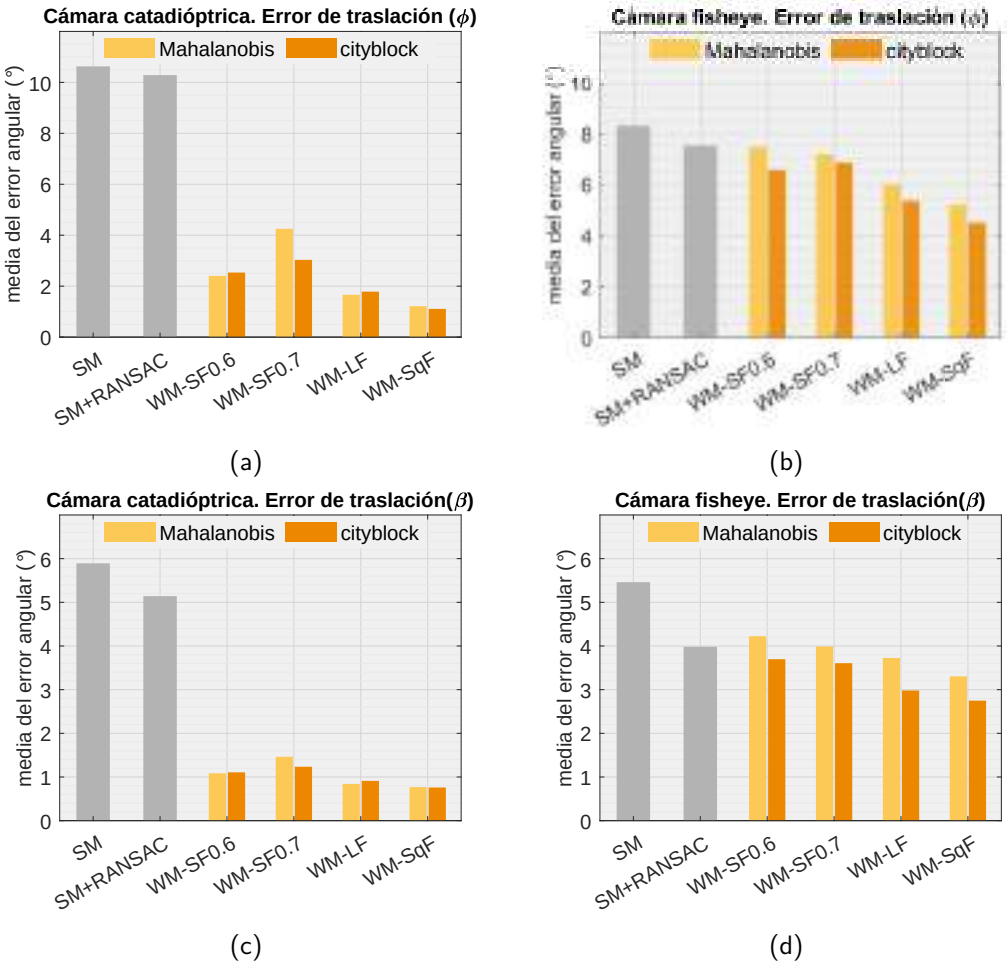


Figura 4.18: Error cometido durante la estimación de la traslación relativa con cada una de las variaciones del método estándar (SM ■) y con el método APOFM ponderado que se muestra con dos barras en función de la distancia para la clasificación de candidatos (Mahalanobis ■ y City-block ■).

similar al método APOFM utilizando la función escalón para la búsqueda de correspondencias ponderada. En general, las variaciones de APOFM presentan una solución más precisa.

En cuanto a la métrica de distancia, observamos comportamientos distintos en función del sistema de visión empleado. Por un lado, los resultados conseguidos con las imágenes *fisheye* tienen una tendencia más clara, pues se logra un menor error en todos los casos cuando la selección de candidatos se basa en kd-tree y City-block. Por otro lado, ante imágenes capturadas por una cámara catadióptrica, no existe una diferencia tan notable si comparamos los resultados con las dos distancias, a excepción del cuarto método (WM-SF0.7) en el que se consigue una estimación más precisa con City-block.

Aunque la traslación es la parte del movimiento que generalmente tiene asociado un mayor error, también es necesario estudiar el error cometido en la estimación de la orientación. Por ello, la figura 4.19 muestra los resultados de este análisis. En cuanto al error al estimar la orientación, el comportamiento de los resultados es muy similar al de la traslación. En el caso de las imágenes *fisheye*, por un lado, se puede ver cómo el error

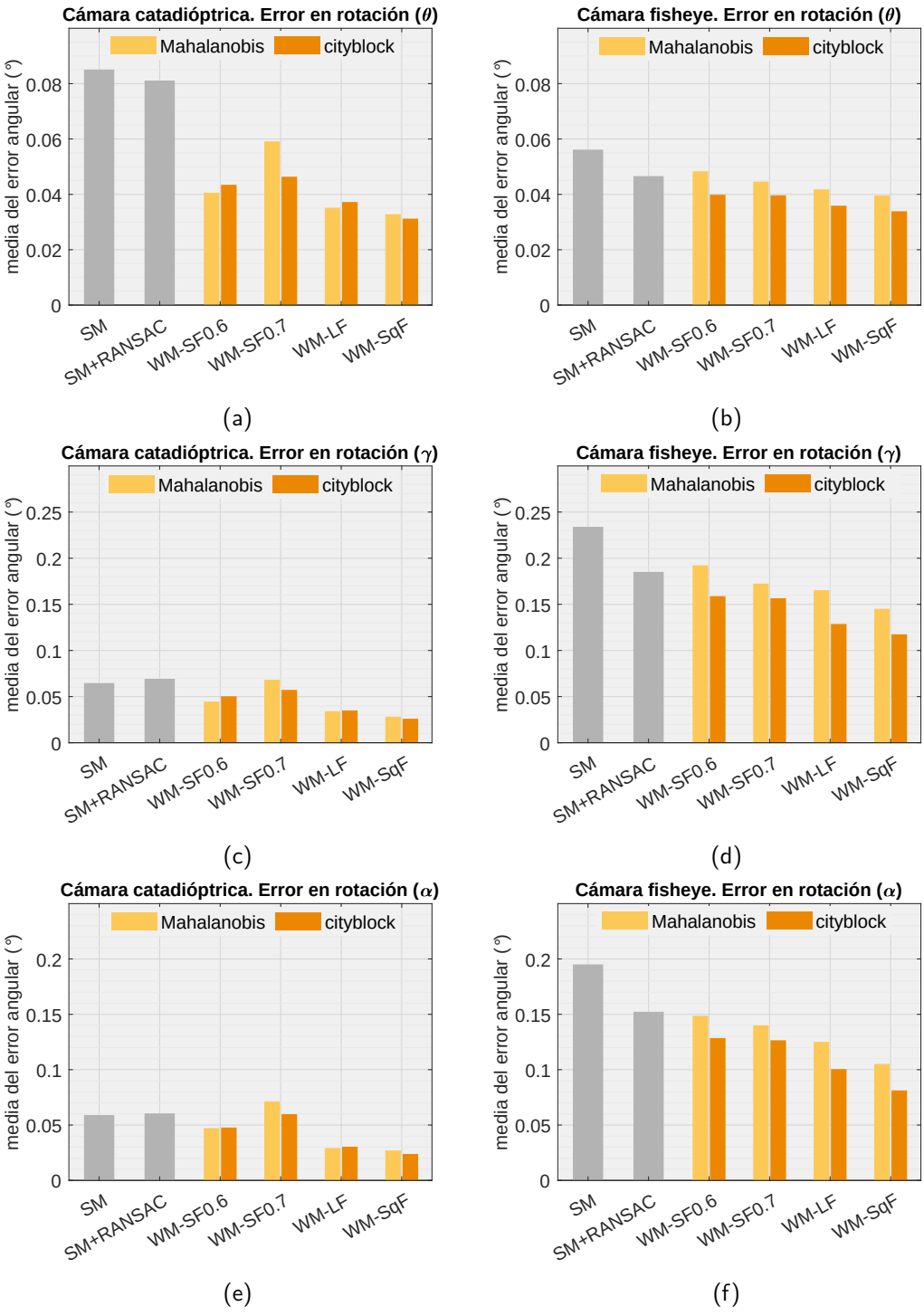


Figura 4.19: Error cometido durante la estimación de la orientación relativa con cada una de las variaciones del método estándar (SM ■) y con el método APOFM ponderado que se muestra con dos barras en función de la distancia para la clasificación de candidatos (Mahalanobis ■ y City-block ■)

es menor al utilizar las funciones lineal y cuadrática, siendo los mejores resultados, en cuanto a precisión, para esta última. Por otro lado, el uso de la distancia de City-block durante la búsqueda de candidatos produce, en todos los casos, un menor error que

con el uso de Mahalanobis. De estas dos conclusiones, la primera sí se cumple para las imágenes catadióptricas, por lo que se puede decir que la búsqueda ponderada mejora la precisión al estimar la orientación con cámaras omnidireccionales. Sin embargo, la segunda conclusión a la que se ha llegado con la cámara *fisheye* no se cumple para la catadióptrica, pues para la búsqueda ponderada con la función lineal y la cuadrática se obtienen, en la estimación de γ y α, resultados muy similares con ambas distancias.

4.5.3 Experimento 2: Evaluación de descriptores locales con el método APOFM

El método APOFM se compone de una etapa en la que se detectan y describen puntos característicos. Por dicho motivo, en este apartado se muestran y analizan los resultados obtenidos tras utilizar el método APOFM con los siguientes tipos de características locales (tabla 4.2): SURF [14], ORB [11], FAST [5] y KAZE [41]. Además, los resultados se comparan con los obtenidos con el método estándar para cada uno de los tipos de características. Como se ha mencionado anteriormente, este experimento se realiza únicamente con el conjunto de imágenes de la base de datos que han sido capturadas con una cámara *fisheye*. Así pues, en este experimento, se realiza un análisis comparativo entre dos de los métodos descritos en la tabla 4.1 (SM y WM-SF0.6) y los tipos de características locales, no entre los sistemas de visión omnidireccionales como en el experimento anterior.

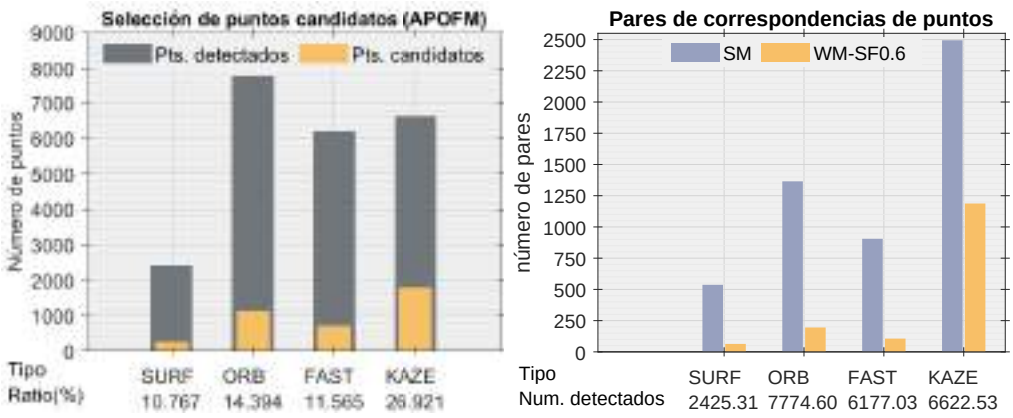
Al igual que el experimento anterior, para la evaluación se tienen en cuenta los siguientes aspectos: (I) número de características coincidentes (apartado 4.5.3.1), (II) número de falsos positivos (apartado 4.5.3.2) y (III) error en la localización (apartado 4.5.3.3). Cabe destacar que, en este experimento se ha utilizado la ponderación mediante función escalón, cuyos parámetros son $p_{min} = 0.6$, es decir se trata del tercer método de la tabla 4.1 (WM-SF0.6), y $(u_{ratio})_{fijo} = 0.6$. Respecto al proceso gaussiano, el paso para crear la rejilla 3D del entorno se ha fijado un valor de $\Delta grid = 0.1$ y el umbral para la determinación de puntos característicos candidatos se ha fijado en $\chi = 0.4463$.

Tabla 4.2: Métodos para la extracción y descripción de características en este experimento.

Método	Característica	Descriptor
SURF [14]	Blobs	Real
FAST [5]	Esquinas	Binario
ORB [11]	Esquinas	Binario
KAZE [41]	Blobs	Real

4.5.3.1 Número de características locales coincidentes

El método APOFM realiza un filtrado de los puntos detectados antes de realizar la búsqueda de características coincidentes. Por dicho motivo, la figura 4.20(a) está dedicada a este método y en ella se puede ver, para cada método de obtención de puntos característicos, el número de puntos característicos detectados y cuántos de ellos se han clasificado como candidatos para buscar su correspondencia. Además, en la parte inferior, se encuentra la ratio entre ambos valores (candidatos/detectados), la cual está expresada en tanto por cien.



(a) (b)

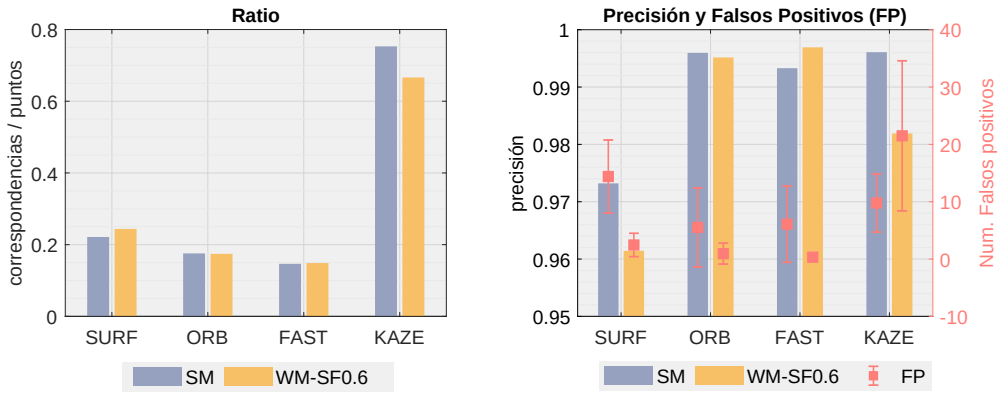
Figura 4.20: Resultados en la búsqueda de correspondencias entre pares de imágenes *fisheye*. En (a) se muestra la cantidad de puntos detectados (■) y cuántos de ellos se han clasificado como candidatos (■). Debajo del eje horizontal se puede ver la ratio entre estos en tanto por cien. En (b) se encuentra representado, en el eje vertical, el número de pares de correspondencias encontradas tanto utilizando el método estándar (■) como WM-SF0.6 (■). Debajo del eje horizontal se muestra el valor medio de número de puntos detectados.

De los resultados podemos ver que, como era de esperar, con el método APOFM el número de puntos característicos candidatos para encontrar correspondencias es menor. Además, con la ratio, se puede ver que el porcentaje de puntos que se consideran candidatos con respecto a los detectados no es constante para todos los tipos de características. El menor valor se ha obtenido con SURF ($ratio_{SURF} = 10.767$) y el más alto con KAZE ($ratio_{KAZE} = 26.921$). Esto quiere decir que se tiene una mayor cantidad de puntos característicos con una probabilidad alta (mayor a 0.6) con KAZE que con SURF.

En cuanto al número de pares de correspondencias encontradas, el valor medio de todos los pares de imágenes para cada tipo de punto característico se representa en el eje Y de la figura 4.20(b). Para cada tipo de punto característico (eje X), la primera barra corresponde a los resultados con el método estándar (SM) mientras que la segunda con el método propuesto (WM-SF0.6). También se muestra debajo de cada tipo, en el eje X, el número de puntos detectados, el cual es el mismo para ambos métodos, SM y WM-SF0.6.

Analizando la figura 4.20(b), observamos que el tipo de características que extrae un menor número de puntos, con diferencia, es SURF. Por el contrario, el que más puntos ha detectado es ORB. Sin embargo, en cuanto al número de pares de correspondencias encontradas, es KAZE el que tiene asociado un valor más alto, seguido de ORB. Al igual que con el número de puntos detectados, SURF es el que menor número de correspondencias ha conseguido, seguido de FAST. Todo esto se cumple tanto para el método estándar como para APOFM, con la diferencia de que el estándar (SM) encuentra más correspondencias, en todos los casos, que el método APOFM (WM-SF0.6).

Para estudiar lo comentado en el párrafo anterior con más detenimiento, la figura 4.21(a) muestra mediante un diagrama de barras en grupos la relación entre el núme-



(a) (b)

Figura 4.21: Se muestra, por un lado (a), la ratio entre el número de puntos que buscan correspondencias y cuántos de ellos finalmente las encuentran. Para APOFM (WM-SF0.6) los puntos que buscan correspondencia están representados por los candidatos. Por otro lado (b), muestra el valor medio de la precisión para cada tipo de punto característico y método. También se representa el valor medio y la desviación (—■—) del número de falsos positivos.

ro de puntos que buscan correspondencias en la otra imagen y cuántos de ellos los encuentran. En el caso del método estándar, que se encuentra en la primera barra de cada grupo, el número de puntos que buscan correspondencias es igual al número de puntos detectados. En el caso del método APOFM, el número de puntos que buscan correspondencias es igual al número de puntos detectados que han sido clasificados como candidatos.

Tras examinar los resultados mostrados en esta figura, vemos que el mayor valor de ratio, con gran diferencia con respecto al resto, corresponde a KAZE. Por el contrario, el menor se ha obtenido con FAST. Además, cabe destacar que, aunque ORB lograba el segundo valor más alto de pares de correspondencias y el primero en número de puntos detectados, la ratio entre ambos valores no es alta, pues incluso SURF está por encima.

4.5.3.2 Falsos positivos

Además de lo estudiado en los apartados anteriores, para poder determinar la robustez, es necesario estudiar el número de falsos positivos o, lo que es lo mismo, cuántos puntos característicos forman un par de correspondencias sin ser la proyección del mismo punto 3D, tal y como ya se ha descrito en apartado 4.4.1.1. El número de falsos positivos obtenidos se muestran, mediante el valor medio y la desviación, en el eje vertical derecho de la figura 4.21(b). Mientras que en el eje vertical izquierdo se representa, mediante un diagrama de barras en grupo, la precisión del paso de búsqueda de correspondencias, pues así también se tiene en cuenta el número de verdaderos positivos. Como sólo conocemos el número de falsos positivos, calculamos la precisión mediante:

$$\text{Precisión} = \frac{N - FP}{N} \tag{4.30}$$

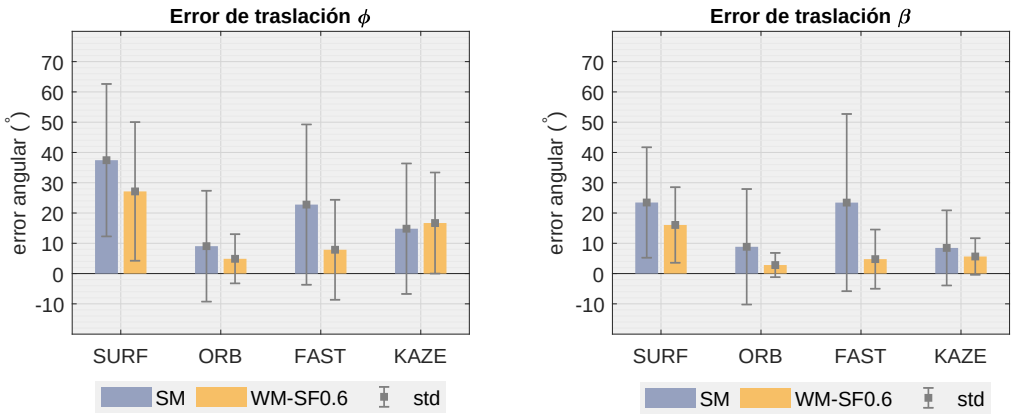
donde N es el número de pares de correspondencias.

A partir de los resultados representados en la figura 4.21(b), podemos concluir que APOFM (WM-SF0.6) presenta con todos los tipos de características, salvo KAZE, los mejores resultados tanto para el valor medio del número de falsos positivos como para su desviación estándar (eje vertical derecho). Así, podemos concluir, que cuando se utilizan puntos SURF, ORB o FAST la robustez del método propuesto es mayor. Para analizar la precisión observamos los resultados representados por las barras en esta misma figura. Teniendo en cuenta que cuanto más cercana a uno sea la altura de la barra, más preciso será el método con respecto a la búsqueda de correspondencia de características, ya que representa el resultado de la ecuación (4.5.3.2). Los resultados muestran que el menos preciso es SURF tanto para el método estándar como para APOFM, presentando este último un valor más pequeño. De hecho, la diferencia con el resto de los tipos de puntos característicos es considerable. En este aspecto, el que mejor resultados logra es el método APOFM con ORB, en lo que respecta a número de falsos positivos y su distribución, pero también a la precisión, siendo esta de 0.995.

4.5.3.3 Error de localización

Al igual que en el primer experimento se ha estudiado el error de localización para cada algoritmo (apartado 4.5.2.3), ahora se va a realizar el mismo estudio, pero para cada uno de los tipos de puntos característicos, y así determinar cuál de ellos consigue una estimación de la pose relativa con mayor precisión.

Por un lado, la figura 4.22 muestra los errores al estimar la traslación relativa entre pares de imágenes consecutivas de la base de datos y, por otro lado, la figura 4.23 (a excepción de la figura 4.23(d)) los obtenidos al estimar la orientación relativa. En todas estas gráficas, el error para cada uno de los parámetros en los que se descompone la pose relativa se visualiza mediante un diagrama de grupos de barras. Cada grupo, corresponde a un tipo de puntos característicos y se compone de dos barras, la primera para el método estándar (SM) y la segunda para el método APOFM. Además, también se muestra mediante una tendencia la desviación estándar en cada caso.



(a) (b)

Figura 4.22: Error angular estimando la traslación, (a) ϕ y (b) β, con imágenes *fisheye*.

La figura 4.22(a) y la figura 4.22(b) representan los errores al estimar los ángulos de traslación, ϕ y β, que se definen en la figura 4.3. Los resultados muestran que

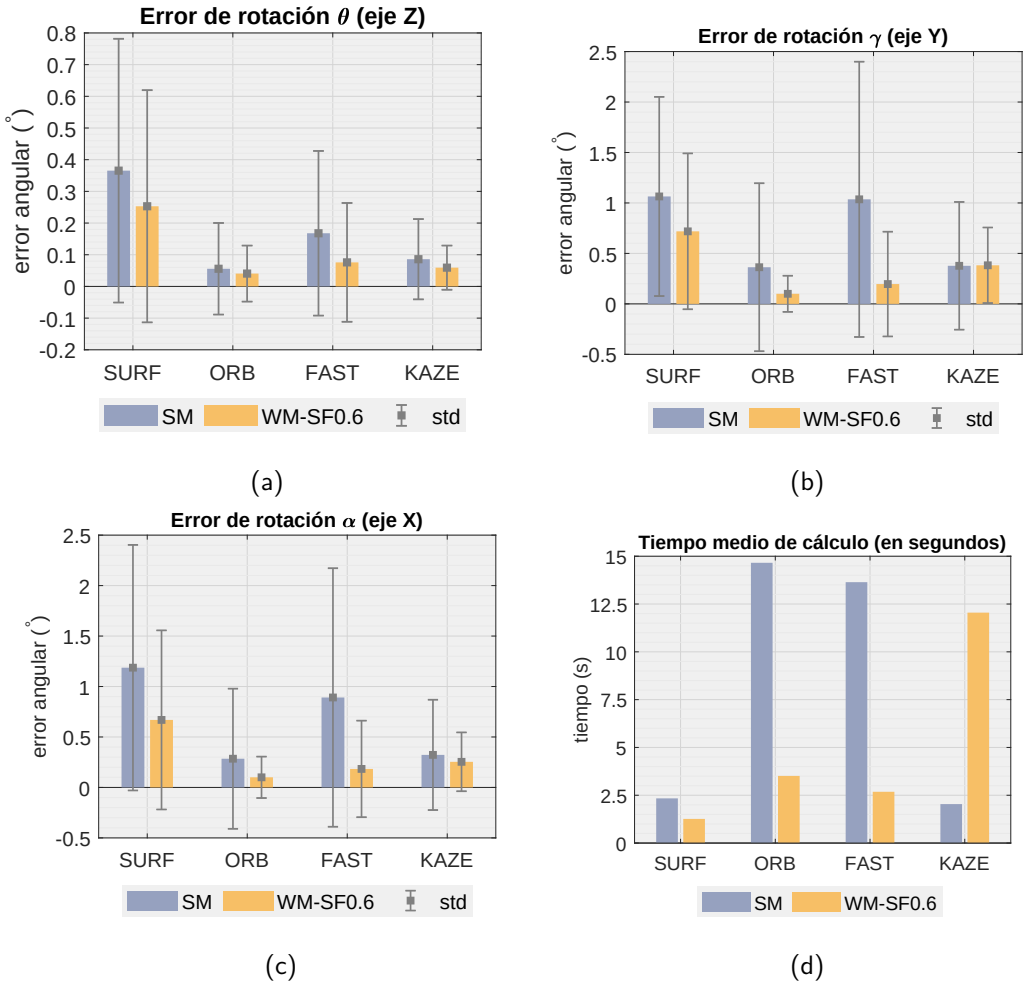


Figura 4.23: Error angular estimando la rotación, (a) θ y (b) γ y (c) α, con imágenes *fisheye*. También se muestra el tiempo de cálculo en (d).

el mayor error cometido se ha producido con SURF. Esto sigue la misma línea que las conclusiones obtenidas acerca de este tipo de punto característico en los análisis anteriores (menor número de puntos detectados y de pares de correspondencias, y también menor precisión). Analizando cada parámetro angular de la traslación por separado, en el caso de ϕ, es decir, de los resultados mostrados en la figura 4.22(a), podemos concluir que el método propuesto (WM-SF0.6) presenta una mejor solución de localización con todos los tipos de puntos característicos, salvo con KAZE, en cuyo caso, el error cometido por ambos métodos es muy similar en términos de valor medio. No obstante, por lo que se refiere a la desviación estándar, APOFM es mejor. En la estimación de β, el método propuesto (WM-SF0.6) presenta los mejores resultados para todos los tipos de puntos característicos.

En cuanto a la rotación, se descompone en tres parámetros angulares, como se muestra en la figura 4.3, θ, γ y α. Cada uno de ellos se analiza de forma independiente en la figura 4.23(a), la figura 4.23(b) y la figura 4.23(c), respectivamente. De este estudio podemos concluir que la combinación de tipo de características y método que mejores resultados han proporcionado a la hora de estimar la pose relativa es ORB

con el método APOFM, analizando conjuntamente los errores cometidos en los cinco parámetros angulares.

4.5.3.4 Tiempo de cálculo

En este experimento también se ha estudiado el tiempo medio de cálculo para todas las combinaciones de tipo de puntos característicos y método de estimación de la pose relativa. Los resultados se muestran en la figura 4.23(d).

Observando el diagrama de barras, podemos determinar que para los tres primeros grupos (SURF, ORB y FAST), la barra que tiene una menor altura es la segunda, es decir, la correspondiente al método APOFM. En estos casos, el método APOFM estima la pose relativa entre un par de imágenes en un menor tiempo que el método estándar, siendo la diferencia considerable para ORB y FAST. Por el contrario, si comparamos las dos barras del cuarto grupo (KAZE) vemos que el tiempo es menor con el método estándar que con el método propuesto y que además la diferencia es bastante notable.

4.6 Conclusión

El presente capítulo ha tenido principalmente dos objetivos. Por un lado, proponer varias contribuciones para mejorar el enfoque APOFM propuesto por Valiente et al. [1], el cual se caracteriza por realizar una búsqueda de correspondencias de puntos característicos basada en un modelo de probabilidad adaptativa acerca de información visual. Por otro lado, extender su aplicación no solo a imágenes capturadas con una cámara catadióptrica sino también con una cámara *fisheye* y realizar un estudio comparativo del desempeño de ambas. Para ello, dicho método se ha implementado en un algoritmo para resolver una tarea muy importante en la robótica móvil, como es el problema de localización a partir de información visual, siendo la cámara catadióptrica y *fisheye* dos sistemas de visión que se usan con frecuencia para dicho fin. Para llevar a cabo los experimentos, se ha utilizado un conjunto de datos disponible públicamente que se compone de imágenes de ambos tipos.

Con respecto al primer objetivo mencionado, hemos propuesto como contribución llevar a cabo una búsqueda de correspondencias ponderada. El modelo de probabilidad en el que se basa el método APOFM [1] se utiliza para determinar uno de los umbrales de la búsqueda de correspondencias, a diferencia del APOFM original [1] en el que dicho modelo únicamente se empleaba para determinar qué puntos característicos podían ser candidatos, seleccionando aquellos con una probabilidad mayor a un cierto umbral (como ocurre al utilizar la función escalón).

Además, por lo que respecta a la clasificación de puntos candidatos, también presentamos otra contribución. Teniendo en cuenta que la clasificación de puntos característicos en candidatos se basa en la distancia a los puntos del modelo proyectados en el plano imagen, proponemos una búsqueda del punto del modelo proyectado más cercano al punto característico detectado mediante un clasificador del k vecino más cercano con una búsqueda basada en el algoritmo kd-tree utilizando la distancia de City-block. Esto es comparado frente a una búsqueda exhaustiva empleando la distancia Mahalanobis. Para finalizar, hemos implementado en el algoritmo un registro de falsos positivos más fiable. Este registro se basa en el hecho de que un par de puntos

de correspondencia representan la proyección de un mismo punto 3D. Así, para cada par de correspondencias, se estima dicho punto 3D y, después de recuperar sus coordenadas, se reproyecta en el plano imagen. El par de correspondencias se etiquetará como falso positivo si el punto 3D reproyectado no se aproxima a la posición del punto característico detectado en dicha imagen.

En términos experimentales, en primer lugar, se ha desarrollado un experimento previo para determinar los valores óptimos de dos parámetros de los cuales depende el método APOFM. Por un lado, el parámetro $\Delta grid$ que determina la resolución espacial del modelo. Por otro lado, se encuentra χ que es el umbral de distancia que permite considerar un punto característico como candidato. Para ello, se ha estudiado la influencia de estos parámetros con respecto al tiempo de ejecución del algoritmo, así como a la precisión en la estimación de los parámetros de movimiento. Ambos valores se han escogido para lograr una solución balanceada entre error y tiempo para ambos sistemas de visión.

A continuación, se ha realizado una comparación del método APOFM con las aportaciones expuestas frente a: (I) un método estándar para estimar el movimiento relativo [303], (II) este mismo algoritmo pero con inclusión de RANSAC para eliminar posibles valores atípicos [239] y (III) el método APOFM original [1] que equivale a los métodos WM-SF0.6 y WM-SF0.7.

Tras analizar los resultados, se ha concluido que el método APOFM con las mejoras propuestas consigue, frente a las variaciones del método estándar, una mayor eficiencia y precisión. Esto se cumple para los dos sistemas de visión omnidireccionales. Centrándonos en las tres variaciones del método APOFM propuestas en este trabajo, los resultados obtenidos corroboran que las aportaciones planteadas mejoran el algoritmo original. Primero, ofrecen ratios más altas entre puntos que buscan correspondencias y los que finalmente los consiguen, siendo el valor más alto de esta proporción el correspondiente al WM-SF0.7. En lo que se refiere a este aspecto, se consiguen un mayor número de candidatos con imágenes tomadas con una cámara *fisheye*. A pesar de esto, el sistema catadióptrico logra un mayor número de correspondencias. Esto se puede deber a que las imágenes tomadas por la cámara *fisheye* presentan un menor campo de visión, además de una mayor distorsión por su naturaleza no lineal.

Además de presentar una mayor ratio, las variaciones del método APOFM presentan un menor número de falsos positivos. Así, se confirma aún más que con las contribuciones de este trabajo se alcanza una mayor precisión y robustez en la búsqueda de correspondencias. Siguiendo con el estudio de falsos positivos, las imágenes *fisheye* son los que mejor responden a este respecto.

También se han evaluado los distintos métodos con respecto al error cometido al estimar el movimiento relativo. En este sentido, hemos observado que se obtiene una mayor precisión al usar una búsqueda de correspondencias ponderada en la que el valor del segundo umbral viene dado por una función cuadrática. Dado que se han evaluado dos posibilidades en la clasificación de puntos candidatos, la que mejores resultados ha conseguido ha sido la búsqueda mediante kd-tree junto con distancia City-block. En cuanto al sistema de visión, se comete menor error al estimar la pose relativa cuando el sistema de visión es el catadióptrico. Es preciso resaltar que se consigue un error menor

a 1° en la estimación de la traslación aun cuando este es el parámetro del movimiento que tiende a presentar mayores errores.

Tras los buenos resultados de estos experimentos, en este capítulo, hemos sugerido realizar una evaluación con distintos puntos característicos y ver cómo influye esta parte del algoritmo en los resultados. Para los experimentos únicamente hemos utilizado una de las variaciones del método APOFM, concretamente WM-SF0.6. Además, se evalúa el comportamiento en imágenes *fisheye* ya que son las que menor precisión de localización han mostrado.

En cuanto a lo mencionado en el párrafo previo, hemos observado que cuando los puntos característicos son extraídos y descritos con SURF, ORB o FAST, el método APOFM ofrece mejores resultados que el método estándar. Por el contrario, esto no sucede con KAZE. Centrándonos en los resultados con APOFM, el menor error medio de localización se ha obtenido con ORB. Además, como se ha visto, al utilizar este tipo de característica local la precisión en la búsqueda de correspondencias es muy cercana a uno, siendo el valor medio de número de falsos positivos bajo y con una desviación pequeña.

Como conclusión, las evaluaciones que se han realizado han demostrado la validez de las aportaciones presentadas. Con ellas se ha podido abordar el problema de localización visual con mayor precisión. En este capítulo, con todos los métodos analizados, se ha estimado el movimiento relativo (a excepción del factor de escala), es decir, la pose relativa viene dada por dos ángulos que corresponden a la traslación y por tres ángulos que definen la orientación. Todos estos experimentos se han realizado con un conjunto de datos de acceso abierto que se componen de imágenes tomadas por dos sistemas de visión que se emplean con frecuencia en la robótica móvil, como son las cámaras catadióptricas y las cámaras *fisheye*. Así ha sido posible realizar una evaluación comparativa en este sentido.

4.6.1 Trabajos futuros

En este capítulo, los experimentos se han realizado con imágenes capturadas en un entorno interior. Por ello, se propone como futuro trabajo ampliar la aplicación de este método para entornos exteriores, ya que, al basarse en un modelo del entorno, el algoritmo tal y como se ha presentado, podría ser sensible y/o mostrar ciertas limitaciones frente a otros factores externos que aparecieran con motivo de trabajar en un entorno no acotado como el de interior. Además, también planteamos evaluar este método con vistas completas de 360° a partir de un sistema de visión compuesto por dos cámaras *fisheye* que apuntan hacia lados opuestos.

Con respecto a la parte de detección y descripción de puntos característicos, planteamos como posible trabajo futuro ampliar dicha evaluación con otras características locales, como sería ASIFT [322] por su invariabilidad ante transformaciones afines, lo cual las puede hacer especialmente relevantes en el caso de imágenes *fisheye* y omnidireccionales.

4.7 Publicaciones en las que se basa este capítulo

Los principales resultados que se han mostrado y discutido en este capítulo se han publicado en:

- M. Flores, D. Valiente, A. Gil, O. Reinoso y L. Payá, "Efficient probability-oriented feature matching using wide field-of-view imaging", Engineering Applications of Artificial Intelligence, vol. 107, pág. 104 539, ene. de 2022, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2021.104539.
  - En este artículo se han presentado varias contribuciones para mejorar el método APOFM. Dicho método consiste en realizar un modelo del entorno con información de probabilidad y aprovechar dicha información para realizar una búsqueda de correspondencias más robusta. El objetivo es utilizar dicho método para resolver la odometría visual. Las contribuciones presentadas en este artículo con respecto al anterior trabajo se enumeran a continuación. En primer lugar, dado que la selección de puntos a encontrar su correspondencia (puntos candidatos) se lleva a cabo buscando el punto del modelo del entorno proyectado en el plano imagen más cercano, se propone realizar dicha búsqueda empleando un clasificador de k vecinos más cercanos con dos métricas de distancia diferentes, cada una con un método de búsqueda específico. En segundo lugar, se propone utilizar el modelo del entorno (distribución de probabilidad espacial) no solo para la selección de puntos candidatos sino también para conseguir una búsqueda ponderada y dinámica de correspondencias de características. En tercer lugar, se utiliza un registro automático de falsos positivos para poder analizar la robustez de los métodos evaluados. Para dicho registro, se calcula el punto 3D para cada par de puntos de correspondencia y se proyecta sobre una de las dos imágenes. Se determinará si dicho par es un falso positivo o no en función de la distancia de dicho punto 3D sobre el plano imagen al punto característico que forma parte del par de correspondencias y que por tanto correspondería a su proyección. En cuarto lugar, se realiza un estudio de la eficiencia de las contribuciones anteriores utilizando no solo un sistema de visión catadióptrico sino también uno compuesto por una lente *fisheye*.
- M. Flores, D. Valiente, S. Cebollada, O. Reinoso y L. Payá, "Evaluating the Influence of Feature Matching on the Performance of Visual Localization with Fisheye Images," en Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics - ICINCO, INSTICC, SciTePress, 2021, págs. 434-441, ISBN: 978-989-758-522-7. DOI: 10.5220/0010555104340441.
  - En este trabajo presentado, el objetivo es analizar la influencia de la parte de procesamiento de la información visual capturada en imágenes *fisheye* en la estimación de la matriz esencial. Para ello, se ha implementado un algoritmo estándar y otro en el que se utiliza el método *Adaptive Probability-Oriented Feature Matching* (APOFM) propuesto por Valiente et al. [1] para la búsqueda de correspondencias. El objetivo de esto es comparar los resultados de ambos algoritmos y además estudiar la influencia de la parte de procesamiento de la información visual en cada uno de estos dos algoritmos.

- M. Flores, D. Valiente, A. Gil, A. Peidró, O. Reinoso y L. Payá', "Evaluación de descriptores locales en localización visual con imágenes ojo de pez", en XLII Jornadas de Automática: libro de actas. Castelló, 1 a 3 de septiembre de 2021, Servizo de Publicacións da UDC, Comité Español de Automática y Universitat Jaume I, ago. de 2021, págs. 507-514, ISBN: 9788497498043. doi: 10.17979/spudc.9788497498043.507.
    - En este trabajo, se han evaluado diferentes tipos de métodos de detección y descripción de puntos característicos locales (como son SURF, FAST, ORB y KAZE) en un algoritmo de estimación de pose relativa entre pares de imágenes tomadas con una cámara *fisheye*. El objetivo de este trabajo es encontrar qué método de extracción de información visual produce un menor error a la hora de estimar la pose relativa con este tipo de imágenes.

5 Generación de una vista completa a partir de un par de imágenes *fisheye*

Cada vez podemos encontrar más cámaras comerciales que capturan imágenes con un campo de visión de hasta 360°. Esto se debe principalmente a que, debido a sus múltiples ventajas, se pueden emplear en muchas aplicaciones, entre las que se encuentra la realidad virtual. Dentro de estas cámaras, la configuración que más se está aplicando consiste en utilizar dos sensores con una lente *fisheye* cada una y que presentan un campo de visión mayor a 180°, posicionados de tal forma que capturen una esfera completa (*back-to-back*). Ahora bien, conseguir una vista esférica completa y de calidad es un desafío. Por dicho motivo, el objetivo principal de este capítulo es intentar generar una vista de este tipo con la mayor calidad posible.

5.1 Introducción

En la actualidad, cada vez se dedican más trabajos de investigación a la creación de imágenes panorámicas, pues presentan la gran ventaja de proporcionar mayor cantidad de información sobre el entorno que las imágenes convencionales. Esta característica es bastante favorable en algunas aplicaciones por lo que su empleo cada vez es mayor en, por ejemplo, la asistencia durante la navegación de sillas de ruedas eléctricas [323-325], la industria del vehículo para proporcionar una vista panorámica [326], realidad virtual y por último, y que más relación tiene con el presente trabajo, es la creación de mapas y localización de robots móviles [327, 328].

Por lo que respecta a la imagen panorámica, se trata de una única imagen que recoge una vista con un gran campo de visión del entorno alrededor del sistema de visión escogido para ello. En este sentido, encontramos diferentes formatos para este tipo de imagen que, representando el entorno como una esfera, pueden ser clasificadas

en atención a si proyectan solamente una porción del entorno (por ejemplo, el formato cilíndrico [329]) o toda la esfera (por ejemplo, el formato equirectangular o del mapa cúbico [330]).

Por cuanto a la adquisición de la imagen se refiere, ciertamente disponemos de diversas alternativas para obtener las mencionadas vistas [331], las cuales se enumeran a continuación. La alternativa más sencilla, en cuanto a elementos ópticos, es girar una única cámara en torno a su centro óptico hasta completar una vuelta completa, es decir, se realizan pequeñas rotaciones respecto a su eje vertical de forma que dos imágenes consecutivas tengan zona de solape. Por otro lado, la segunda alternativa se basa en la idea anterior, pero, en vez de tener que girar la cámara, se utiliza un conjunto de cámaras apuntando hacia distintas direcciones. La última alternativa consiste en combinar una cámara con lentes de gran angular o espejos. En el caso de las dos primeras alternativas, como se tiene más de una imagen, estas deben fusionarse posteriormente para así tener la imagen panorámica.

Respecto a la última alternativa, el sistema de visión como resultado de dicha combinación no cubre todo el entorno. Sin embargo, esto se puede lograr si, en vez de un único sensor, se utilizan dos sensores, cada uno de ellos con una lente *fisheye* de forma que su campo de visión sea superior a 180°. Para poder capturar la esfera completa, es decir, lograr obtener un campo de visión de 360° horizontalmente y 180° verticalmente, el punto de vista de un sensor y su lente *fisheye* debe ser opuesto al del otro sensor y su respectiva lente *fisheye*, configuración conocida como *back-to-back*. De hecho, esta configuración es la utilizada en algunas cámaras comerciales como la Samsung Gear 360 [332], la RICOH THETA S [333] o la Garmin VIRB 360 [262]. Si bien las dos primeras se han empleado en varios trabajos de investigación [334-336] y por tanto, sus características han sido más profundamente estudiadas, el uso de la tercera de ellas, la Garmin VIRB 360, no es tan extensa en este ámbito por lo que ha sido la seleccionada para este trabajo.

De este modo, las cámaras cuya configuración se compone de dos lentes *fisheye* opuestas, capturan en el mismo disparo dos imágenes *fisheye*, así que, si se quiere una única imagen, dicho par debe procesarse para finalmente combinarse y lograr una imagen completa de alta resolución. Asimismo, cabe destacar, como ventajas, que estas cámaras son ligeras, pequeñas y con un coste asequible. Sin embargo, no se puede obviar que la tarea descrita supone un importante reto debido a las características que presentan. En primer lugar, los centros de proyección de las citadas lentes se encuentran desplazados, es decir, existe paralaje, lo cual genera desajustes en relación con las características coincidentes y puede producir un efecto fantasma en la zona de solape. En segundo lugar, la zona más alejada del centro de este tipo de imagen y, por ende con mayor distorsión, corresponde a la zona de solape, lo que conlleva que el emparejamiento entre ellas no sea directo y requiera de un procesamiento previo. En tercer lugar, las imágenes tienen un campo de visión superpuesto limitado, lo que conlleva que no se pueda extraer demasiada información de la zona de solape. Además, hay que tener en cuenta que, como ya se ha mencionado anteriormente, esta última es la región más afectada por la distorsión.

Por todo lo anterior, muchos trabajos de investigación se centran en buscar soluciones a los problemas descritos, a fin de alcanzar vistas completas de alta calidad. En

general, los algoritmos que, a partir de un par de imágenes *fisheye*, generan una vista esférica completa, suelen comenzar con una transformación de las imágenes ojo de pez a un formato esférico (para solucionar el segundo reto comentado). Después, realizar una alineación posterior de este par de imágenes equirectangulares y, por último, combinar estas dos en una única vista esférica completa. Para la fusión de ambas, se emplea alguna de las técnicas propuestas para dicha finalidad, pues estas están diseñadas para suprimir las posibles incoherencias existentes. En el apartado 5.2 se verán algunas de las soluciones propuestas en la literatura, centrándonos en las correspondientes a los dos primeros pasos del algoritmo descrito.

Aunque estos son los tres pasos indispensables para conseguir una vista completa a partir de un par de imágenes *fisheye*, en algunos casos añaden pasos adicionales, como un proceso de calibración o una compensación fotométrica, para adquirir una vista completa con mayor calidad.

5.1.1 Contribuciones de este capítulo

El propósito del presente capítulo es generar una vista esférica completa a partir de un par de imágenes *fisheye* con la mayor calidad posible. Si bien la cámara escogida para adquirir el par de imágenes *fisheye* ya tiene implementado un algoritmo que proporciona de forma directa este tipo de vista, se observa que, en numerosas ocasiones, la calidad no es todo lo adecuada que sería deseable. Esto ocurre especialmente cuando la zona de solape es rica en textura. Por este motivo surge la necesidad de abordar el problema de generación de una vista esférica completa de calidad.

Para dicho propósito, como se verá en apartados posteriores, se sugiere un conjunto de variaciones del algoritmo propuesto. Dichas variaciones se centran en: (I) la proyección para mapear puntos desde la esfera a la imagen *fisheye* al representar estas últimas en formato esférico y (II) el tipo de transformación que se estima durante el proceso de registro del par de imágenes esféricas.

Respecto a las contribuciones de este capítulo, se enumeran a continuación:

1. Se ha realizado un estudio sobre la relación entre pares de correspondencias de puntos encontrados en pares de imágenes equirectangulares y también en pares de imágenes *fisheye*. De dicho estudio surgieron las dos aportaciones siguientes.
2. En la transformación a formato esférico de la imagen *fisheye* trasera, se propone un paso de corrección de las coordenadas polares basado en la relación obtenida en el estudio comentado en la aportación anterior (entre el par de imágenes *fisheye*).
3. Se propone estimar y aplicar durante el proceso de registro una transformación que corresponde a un polinomio de grado dos.
4. Para la evaluación de la calidad de la zona de solape de las vistas completas, se propone un algoritmo que requiere de una imagen de referencia. Este algoritmo consta de dos arquitecturas de redes convolucionales idénticas y con mismos pesos, donde una de ellas extrae un descriptor holístico para la zona de solape a evaluar y la otra para la zona definida como referencia. La medida que determina la calidad relativa de la primera corresponde a la distancia de correlación entre

ambos descriptores.

5. Adicionalmente, también se propone otro procedimiento de evaluación en función del número de marcas ArUco que son reconocidas.
6. Se ha evaluado y comparado la calidad de las vistas completas generadas con las diferentes opciones que se plantean a lo largo del capítulo y la proporcionada de forma directa por la cámara Garmin VIRB 360. Para ello, se han empleado diferentes medidas de calidad con el fin de realizar una evaluación completa.

5.2 Trabajos relacionados

Hay que tener en cuenta que, hoy en día, la información visual se usa para resolver diversas tareas. Por ejemplo, Zichao Zhang et al. [39] llevan a cabo una evaluación de tres sistemas de visión con distinto campo de visión para resolver el problema de localización. De dicho trabajo extrajeron, como conclusión, que el uso de imágenes que han sido capturadas por cámaras con un gran campo de visión mejora la localización en entornos de interior.

Por otro lado, cabe destacar que pueden emplearse diferentes sistemas de visión para obtener una imagen panorámica y que pueden clasificarse en función de, por un lado, el número de cámaras convencionales que utilicen y, por otro, si se combinan o no con otro elemento (como lentes *fisheye* o espejos).

En el caso de que el sistema de visión esté compuesto por más de una cámara (o sensor), estas imágenes capturadas se tienen que combinar para conseguir una sola, esto se puede lograr con técnicas de *stitching*. Lyu et al. [337] presentan un estudio de las técnicas de *stitching* de imágenes para construir una imagen panorámica. En esta misma línea, Szeliski [141] también presenta en [141] una revisión acerca de esta técnica.

En cuanto a la primera tipología referida (en atención al número de cámaras), en primer lugar, para la tarea descrita podemos utilizar una sola cámara, pero es preciso que la misma realice distintas capturas, realizando, por ejemplo, una serie de rotaciones alrededor del eje vertical para cubrir todo el entorno, como se ha comentado en el apartado anterior, y así generar una secuencia de imágenes con la que generar la imagen panorámica. En este caso, el problema que se plantea es la imposibilidad de capturar todas las imágenes en un mismo instante de tiempo (o disparo), lo cual presenta un inconveniente importante en relación con las aplicaciones en las que la cámara tiene un movimiento continuo, como la navegación de robots móviles. Igualmente, este hecho puede causar algunas dificultades en el proceso de *stitching*, especialmente si el entorno es dinámico. En [338], se describen algunos de estos sistemas panorámicos y se propone un sistema automático panorámico cuyo control se realiza a través de un *smartphone*.

En segundo lugar, por lo que respecta a los sistemas de visión compuestos por varias cámaras que apuntan en distintas direcciones con campos de visión superpuestos, constituyen una alternativa que solventa la problemática comentada sobre el sistema anterior. En este caso, se necesitan varias cámaras para obtener una vista completa de 360°, lo que da lugar a muchas zonas en la imagen en las que pueden producirse efectos derivados del proceso de *stitching*. Una posible solución para reducir el número

de cámaras, y con ello las imágenes a procesar, consiste en combinar las cámaras (o al menos alguna de ellas) con lentes de gran angular tales como *fisheye*. Por ejemplo, esto lo podemos ver en las cámaras comerciales: surround360 de Facebook e Insta360 PRO. La primera de ellas, la surround360 de Facebook, está compuesta por 17 lentes, 14 de gran angular y 3 *fisheye*, mientras que la segunda, la Insta360 PRO, tiene seis lentes *fisheye*. Zhang et al. [339] propusieron un algoritmo basado en el flujo óptico para generar una imagen panorámica y utilizaron imágenes capturadas por las dos cámaras comentadas.

En el párrafo anterior se ha comentado que es posible aumentar el campo de visión de una sola imagen combinando la cámara convencional con una lente *fisheye*, pero no es la única combinación posible, pues esto también se logra mediante una superficie reflectante (sistema de visión catadióptrico). La combinación de una cámara convencional con uno de estos dos tipos de elementos es un sistema de visión cuyo uso está relativamente extendido en aplicaciones robóticas. Por ejemplo, Flores et al. [340] evalúan un método probabilístico de búsqueda de correspondencias entre pares de imágenes. Para ello, lo implementan en un algoritmo para la estimación de la pose relativa entre dos instantes de tiempo y utilizan imágenes capturadas por estas dos configuraciones. Asimismo, Cabrera et al. [170] propusieron estimar los hiperparámetros óptimos de una Red Neuronal Convolucional (CNN, acrónimo del inglés *Convolutional Neural Network*), que se emplea para abordar el problema de localización de robots móviles utilizando imágenes capturadas por un sistema de visión catadióptrico.

Un sistema de visión catadióptrico tiene la ventaja de producir una única vista del entorno, que se puede transformar conjuntamente para obtener su proyección panorámica, lo que significa que dicha imagen no sufre los efectos derivados del proceso de *stitching*, pues no es necesario implementar esta técnica. Sin embargo, lo habitual es que se obtenga una imagen con una resolución inferior. Además, capta un campo de visión inferior a una esfera entera (se excluyen la parte superior e inferior de la esfera, es decir, las zonas cercanas a los polos).

En cuanto a las lentes *fisheye*, son capaces de capturar un campo de visión algo mayor de una semiesfera y suelen emplearse para tareas de vigilancia visual. A este respecto, Wang et al. [341] realizaron un estudio basado en la detección de personas para vigilancia utilizando Mask-RCNN en imágenes tomadas por una cámara *fisheye* de vista en planta. En cambio, se pueden utilizar dos lentes *fisheye* con un campo de visión superior a 180° cada una y direcciones de visión opuestas para conseguir la esfera completa con la utilización de dos cámaras solamente.

En los últimos años, las cámaras compuestas por dos lentes *fisheye*, posicionadas *back-to-back*, están ganando interés debido a las numerosas ventajas que presentan. No obstante, como se ha comentado en el apartado anterior, el reto que plantean estas cámaras es generar una vista de 360° de alta calidad, es decir, sin sufrir los efectos producidos por el proceso de *stitching*: efecto fantasma, desalineación, distorsión estructural, error geométrico, aberraciones cromáticas y desenfoque. Tian et al. [342] describen cada uno de estos efectos, su origen y sus principales características.

En cuanto a esta última configuración para adquirir una vista completa del entorno, existen diversos trabajos de investigación sobre el estado del arte que han estudiado

métodos para obtener vistas de 360° de alta calidad a partir de dos lentes *fisheye*.

Como ya se ha hecho referencia en el apartado anterior, las principales etapas de estos métodos son la transformación de las imágenes *fisheye* a formato esférico (*unwarping*), el proceso de registro y el proceso de fusión. En los párrafos siguientes se describen los principales enfoques, centrándose en las alternativas para abordar estas etapas de la forma más óptima posible.

Ho y Budagavi [150] proponen un algoritmo en el que el par de imágenes *fisheye* son proyectadas a formato equirectangular y posteriormente el proceso de registro de dicho par se realiza en dos pasos. En el primero, dado un conjunto de puntos de control seleccionados manualmente, se estima la matriz afín para minimizar el desajuste geométrico entre ambas imágenes equirectangulares. En el segundo paso, se lleva a cabo una correspondencia de plantillas para los objetos de la zona de solapamiento. Sin embargo, este método solo produce una alineación parcialmente precisa. Esto se debe a que los puntos de control en la parte central de la vista esférica equirectangular se alinean bien, pero esto no sucede cuando dichos puntos se encuentran en las partes superior o inferior. Por este motivo, los autores proponen un método mejorado en [151]. La mejora consiste en no estimar la matriz afín sino en realizar la transformación mediante interpolación de rejilla basada en mínimos cuadrados móviles rígidos (MLS, del inglés *rigid Moving Least Squares*). Además, en este trabajo los autores también amplían el trabajo anterior con respecto al formato de los datos de entrada para que se pueda utilizar con vídeos.

Por otro lado, Lo et al. [343] sugieren realizar la alineación del par de imágenes mediante una deformación local de malla, la cual se basa en la minimización de una función de coste que combina dos términos. Antes de describir ambos términos, cabe destacar que una vez transformadas las imágenes *fisheye* a formato equirectangular, se dividen en una malla uniforme. En lo que respecta a la función de coste, por un lado, el primer término, denominado de característica, tiene como objetivo alinear cada par de puntos característicos de correspondencia lo más cerca posible de su punto central. Por otro lado, el segundo, denominado término de suavidad, trata de preservar la estructura geométrica de la malla.

Souza et al. [152] proponen un método de *stitching* adaptativo basado en regiones de imagen de alta textura. Cuando las dos imágenes *fisheye* se convierten en proyecciones equirectangulares, se extraen características ORB de las regiones solapadas y estas se agrupan en plantillas. A continuación, los autores intentan minimizar la discontinuidad mediante un ajuste de plantillas utilizando solo las plantillas obtenidas en el paso anterior (zonas de alta textura) en lugar de todas las regiones solapadas. Tras ello, la información de desplazamiento obtenida a partir del ajuste de plantillas se utiliza para estimar la matriz de homografía.

Por su parte, Xue et al. [344] se centran en tratar de mejorar dos etapas del algoritmo. Por un lado, utilizan la interpolación bilineal de latitud y longitud (LLBI, acrónimo del inglés *Latitude Longitude Bilinear Interpolation*) durante la transformación de las imágenes *fisheye* a rectangulares. Por otro lado, los autores proponen una mezcla ponderada empleando una función no lineal, concretamente, la función arco.

Ni et al. [153] proponen un algoritmo para alcanzar, fundamentalmente, dos objeti-

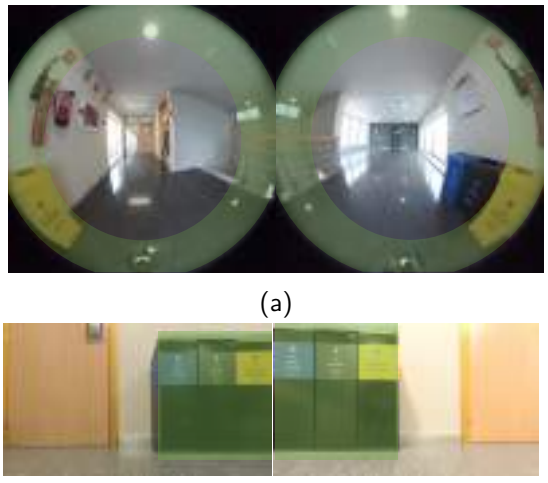
vos. Por un lado, corregir la distorsión generada por la lente *fisheye* (distorsión f-theta) y, por otro lado, eliminar el error de instalación. Para lograr tal cometido, en primer lugar, durante el *unwarping* de las imágenes *fisheye*, utilizan un método de interpolación lineal para relacionar los dos parámetros (r y θ) de la proyección equidistante de *fisheye*. Además, estiman el centro y el radio del área efectiva de la imagen ojo de pez. En segundo lugar, los autores proponen estimar una matriz de rotación para relacionar las coordenadas espaciales de las esferas correspondientes a la *fisheye* delantera y a la trasera.

En otro orden de cosas, Lo et al. [345] realizan una calibración de una cámara con dos lentes *fisheye*. Así, plantean una calibración concéntrica, para lo cual se estiman conjuntamente los centros ópticos y los parámetros de las funciones de proyección (coeficientes polinómicos) correspondientes al par de lentes *fisheye*. Además, proponen la aplicación de una alineación local basada en la deformación de la malla.

Por último, Lin et al. [346] desarrollan un método para estimar el radio efectivo de una imagen *fisheye*, así como el campo de visión efectivo. Tras ello, estos parámetros estimados se utilizan para transformar las imágenes *fisheye* a vista esférica equirectangular.

5.3 Generación de una vista completa a partir de un par de imágenes *fisheye*

Las cámaras compuestas por dos lentes *fisheye* con un campo de visión mayor a 180° colocadas de forma que sus direcciones de vista son opuestas, una apuntando hacia el frente y la otra hacia atrás, son capaces de proporcionar un par de imágenes a partir de las cuales se puede conseguir una única vista completa del entorno.



(a)

(b)

Figura 5.1: Zonas de solape (señaladas en verde) entre imágenes de distintos formatos (a) par *fisheye* y (b) dos que siguen el modelo pinhole.

La combinación de varias imágenes en una sola se logra principalmente aplicando dos técnicas de manera consecutiva: el registro de imágenes y la fusión (también conocida como *blending*). Estos dos pasos suelen ser, en general, suficientes con imágenes

que siguen el modelo de proyección pinhole. Sin embargo, cuando las imágenes a partir de las cuales se va a generar la vista de 360° son *fisheye*, no basta con esto debido a la forma en la que se captura el entorno. En este caso, las imágenes tienen una forma circular y la información común se encuentra en la zona más alejada del centro de la imagen y con mayor distorsión. En otras palabras, la zona de solape tiene forma de corona circular. Por dicho motivo, estas imágenes deben expresarse en otro formato con el que el proceso de combinación sea más fácil e intuitivo. Esto se muestra en la figura 5.1, donde se puede ver la zona de solape de un par de imágenes *fisheye* (figura 5.1(a)) y la zona de solape de un par de imágenes pinhole (5.1(b)).

Teniendo en cuenta que las imágenes *fisheye* pueden proyectarse sobre una esfera unitaria, estas pueden representarse en un formato esférico (por lo comentado en el párrafo anterior) utilizando un mapeo de esfera a plano, como la proyección equirectangular o la proyección cúbica (seis planos). El tipo de representación más empleada para vistas de 360° es la proyección equirectangular [347-349].

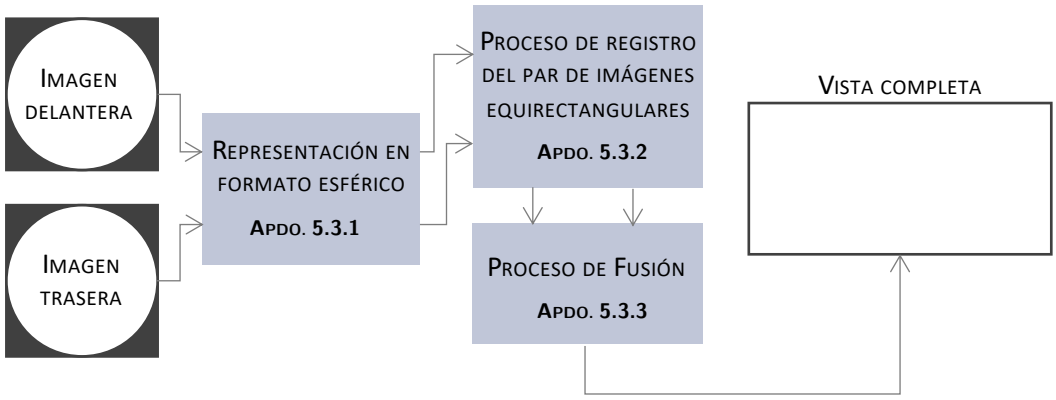


Figura 5.2: Diagrama de bloques con los tres módulos principales para la generación de una vista completa a partir de un par de imágenes *fisheye*: representación en formato esférico (apartado 5.3.1), proceso de registro (apartado 5.3.2) del par de imágenes equirectangulares obtenidas en el módulo anterior y proceso de fusión (apartado 5.3.3) en una única imagen que corresponde a la vista esférica completa.

Así, según lo expuesto en este apartado, el algoritmo para generar una vista de 360° se compone principalmente de tres etapas (o módulos), como muestra el diagrama de bloques de la figura 5.2: representación en formato esférico (apartado 5.3.1), proceso de registro (apartado 5.3.2) y proceso de fusión (apartado 5.3.3).

5.3.1 Representación en formato esférico

Por la forma en la que una cámara *fisheye* captura el entorno, esta puede representarse como una esfera unitaria donde cada punto 3D es proyectado sobre ella antes de ser proyectados sobre el plano imagen. Teniendo en cuenta esto, dada una imagen *fisheye*, cada píxel puede proyectarse sobre una esfera unitaria. Después solo se tendría que mapear dichos puntos ubicados sobre la esfera a un plano 2D mediante alguna proyección esférica como la equirectangular.

De esta forma, este módulo del algoritmo se compone de dos proyecciones, un primer mapeo desde la imagen ojo de pez a la esfera unitaria y un segundo mapeo desde la esfera unitaria al plano rectangular usando la proyección esférica equirectangular. Tras estas dos proyecciones, la imagen *fisheye* es representada como una vista esférica. Hay que tener en cuenta que, dado que una única imagen *fisheye* no está capturando toda la escena, la imagen esférica resultante de esta transformación tendrá zonas en las que el valor de los píxeles será nulo, pues esa zona de la esfera no ha sido capturada por dicha lente.

Siguiendo con lo comentado al inicio del párrafo anterior sobre el primer módulo del algoritmo, se necesitan dos funciones de proyección. Por un lado, una proyección para el mapeo desde la imagen *fisheye* a la esfera unitaria (y viceversa). Por otro lado, otra proyección para el mapeo desde la esfera unitaria al plano imagen esférica (y viceversa). Para el primero, se han implementado dos alternativas: (I) utilizando un modelo de cámara basado en esfera, concretamente el propuesto por Scaramuzza et al. [10] o (II) una función de proyección *fisheye* como es la equidistante. Para el segundo mapeo, de la esfera al plano imagen esférica, se ha empleado una proyección equirectangular. Este tipo de proyección se caracteriza por representar una esfera en una imagen con una relación de aspecto de 2:1, ya que se estaría representando 360° en el eje horizontal y 180° en el vertical. Además, la información visual que se encuentre en el polo norte/sur de la esfera aparecerá en la parte superior/inferior de la imagen esférica, respectivamente.

El orden que se ha descrito hasta ahora corresponde al mapeo directo, es decir, para cada píxel de la imagen *fisheye* (origen) se obtiene su correspondiente píxel en la imagen esférica (destino). Sin embargo, para asegurarnos de que la imagen esférica no tenga ningún píxel vacío (a dicho píxel no se le ha asignado ninguno de la imagen de origen), realizamos la transformación mediante un mapeo hacia atrás (inverso).

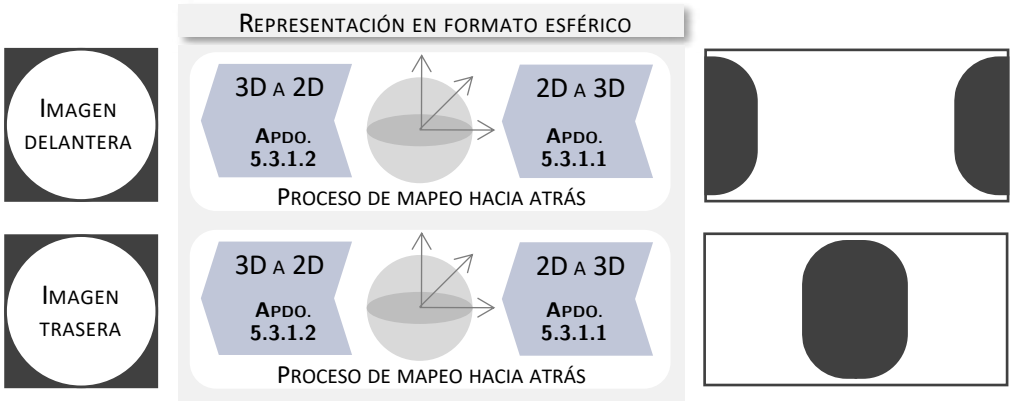


Figura 5.3: Diagrama de bloques para la representación de cada imagen *fisheye* en formato esférico mediante un proceso de mapeo hacia atrás.

De este modo, en esta etapa del algoritmo, se transforma la imagen *fisheye* a un formato esférico iterando sobre cada píxel de la imagen destino (imagen esférica) que se le aplica la transformación inversa para obtener su posición en la imagen de entrada (imagen *fisheye*). Esta transformación inversa se compone de un mapeo desde

la imagen esférica a la esfera unitaria (apartado 5.3.1.2) seguido de otro mapeo desde la esfera a la imagen *fisheye* (apartado 5.3.1.1), esto se puede visualizar en la figura 5.3.

5.3.1.1 Mapeo desde imagen esférica a esfera unitaria: 2D a 3D

Antes de empezar a aplicar las transformaciones, las coordenadas de la imagen esférica (u_{out}, v_{out}) deben ser normalizadas y referidas con respecto al centro de esta imagen.

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = \left(\begin{bmatrix} a_x & 0 \\ 0 & a_y \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \right)^{-1} \cdot \left(\begin{bmatrix} u_{out} \\ v_{out} \end{bmatrix} - \begin{bmatrix} c_x \\ c_y \end{bmatrix} \right) \tag{5.1}$$

donde a_x y a_y corresponden a la mitad de la anchura y la altura, respectivamente, de la imagen esférica; mientras que c_x y c_y son las coordenadas del centro de la imagen esférica.

Como ya se ha mencionado, para mapear un punto desde el plano imagen esférica a la esfera se utiliza la proyección equirectangular, la cual establece una relación proporcional entre la coordenada horizontal (x_{out}) y la longitud (θ) de la proyección sobre la esfera unitaria. De igual forma, para la coordenada vertical (y_{out}) y la latitud (α). Así, partiendo de la base de que la longitud (θ) se encuentra dentro del rango $[-\pi, \pi]$ y la latitud (α) en $[-\pi/2, \pi/2]$, podemos expresar las coordenadas cartesianas (x_{out}, y_{out}) en ángulos (θ, α) mediante las siguientes ecuaciones:

$$\alpha = \pi/2 \cdot y_{out} \tag{5.2}$$

$$\theta = \pi \cdot x_{out} \tag{5.3}$$

Como la esfera es unitaria, solo con estos dos ángulos se pueden calcular las coordenadas en el espacio 3D de la siguiente forma:

$$\boldsymbol{P_G} = \begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \begin{bmatrix} \cos\alpha \cos\theta \\ \cos\alpha \sin\theta \\ \sin\alpha \end{bmatrix} \tag{5.4}$$

5.3.1.2 Mapeo desde esfera unitaria a imagen *fisheye*: 3D a 2D

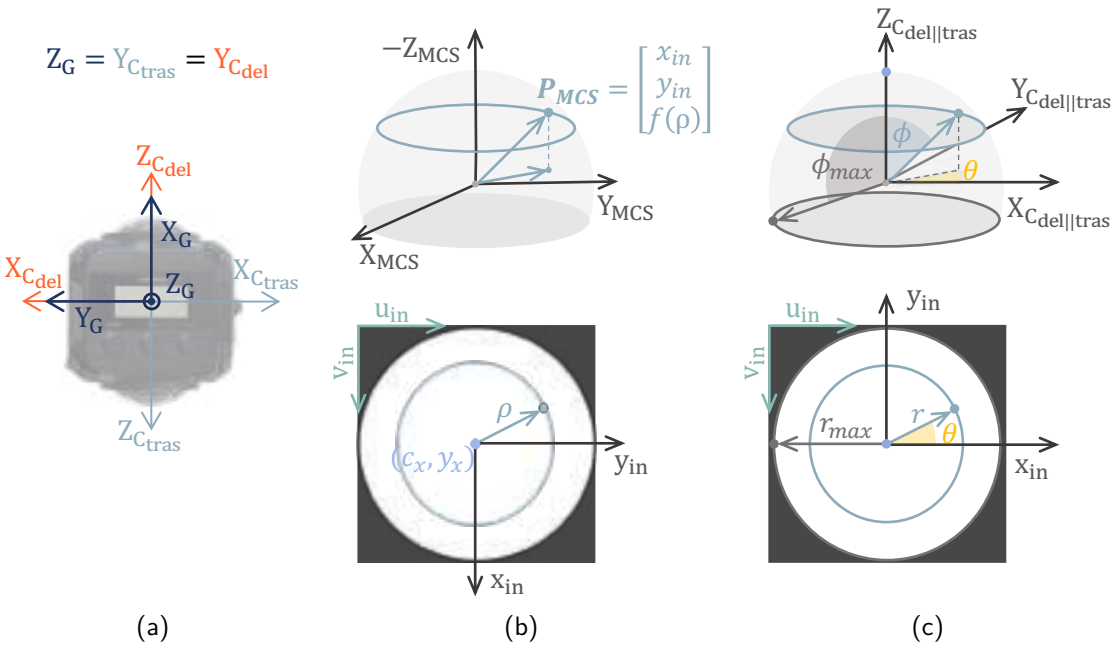
Una vez se tienen las coordenadas sobre la esfera unitaria, se realiza un segundo mapeo para proyectar todos estos puntos 3D sobre la imagen *fisheye*. Para esto, se han implementado dos alternativas que serán evaluadas y comparadas en cuanto a la calidad de la vista esférica total que se genera. La primera opción para proyectar sobre el plano imagen *fisheye* es el modelo de cámara esférica propuesto por Scaramuzza et al. [10] (apartado 5.3.1.2.1). En este caso, se debe realizar previamente un proceso de calibración para estimar los parámetros de este modelo tanto para la cámara delantera como para la trasera. La segunda opción se basa en aplicar la proyección equidistante (apartado 5.3.1.2.2).

Antes de proceder al segundo mapeo, es necesario realizar un cambio de sistema de referencia. Esto se debe a que las coordenadas 3D obtenidas en el mapeo anterior están

referidas a un sistema de referencia ($\{\mathrm{G}\}$) en el que el eje Z apunta hacia arriba (polo norte), es decir, es perpendicular al plano del suelo. Sin embargo, asumimos que la cámara *dual-fisheye* se encuentra colocada de tal forma que el eje Z frontal y el trasero son colineales en un plano paralelo al del suelo y, por el contrario, el eje perpendicular a este último es el eje Y. Esto se muestra gráficamente en la figura 5.4(a). Además, también asumimos que el centro de la esfera en la que se ha proyectado y el centro de la cámara delantera y trasera son el mismo, es decir, no existe translación entre estos tres sistemas de referencia.

Teniendo en cuenta todo lo mencionado en el párrafo anterior, la transformación entre el sistema de coordenadas globales y el de la cámara delantera/trasera consiste en una rotación y dependerá de la imagen *fisheye* que se esté procesando. Tanto la transformación para la cámara delantera como para la trasera desde el sistema de coordenadas global se compone de dos rotaciones, una sobre el eje $\mathrm{Z_G}$ seguida de otra sobre el eje $\mathrm{X_G}$. En el caso de la imagen frontal ($\mathbf{R_{GC_{del}}}$), la primera rotación descrita es de $90°$ al igual que la segunda. Por el contrario, en el caso de la imagen trasera ($\mathbf{R_{GC_{tras}}}$), la primera rotación es de $-90°$ y la segunda coincide con la frontal (de $90°$).

Una vez aplicada la transformación correspondiente ($\mathbf{R_{GC_{del}}}$ o $\mathbf{R_{GC_{tras}}}$) a las proyecciones 3D, estas se encuentran referidas al sistema de coordenadas de la cámara y, para continuar con el proceso, se aplica uno de los dos tipos de proyecciones.



(a) (b) (c)

Figura 5.4: En (a) se muestran los tres sistemas de referencia con los que se trabaja durante la transformación a formato esférico. Para llevar a cabo esto último se realiza un mapeo de puntos desde la esfera al plano imagen *fisheye* para lo cual se necesita de un modelo de proyección: (b) modelo de cámara de Scaramuzza et al. [10] o (c) proyección equidistante.

5.3.1.2.1. Proyección mediante un modelo de cámara esférica. Como ya se ha mencionado, el modelo de cámara escogido es el propuesto por Scaramuzza et al.

[10]. En dicho modelo, la función de proyección $g(x, y)$ se basa en un polinomio de Taylor ($f(\rho)$) que depende de la distancia al centro (ρ) y cuyos coeficientes y grado se estiman durante el proceso de calibración. Además, como se verá más tarde, la proyección de 3D a 2D no se realiza directamente sobre el plano imagen, sino sobre un plano hipotético, por lo que x e y serán las coordenadas en dicho plano ideal y a partir de ahora se identificarán mediante x_{in} e y_{in}.

A continuación, se muestran las ecuaciones para proyectar un punto 3D sobre la imagen *fisheye*, aunque es necesario mencionar que se debe realizar otra transformación de sistemas de coordenadas para poder aplicarlas. En este caso, cada punto 3D obtenido tras aplicar $\mathbf{R}_{\mathbf{GC}_{\mathbf{del}}}$ o $\mathbf{R}_{\mathbf{GC}_{\mathbf{tras}}}$ es multiplicado por la matriz de rotación que se deriva de realizar un giro de $90°$ sobre el eje $\mathrm{Z}_{\mathrm{C}_{\mathrm{del}\|\mathrm{tras}}}$. Además de esto, después se invierte el signo de la coordenada Z.

Tras expresar el punto 3D en el sistema de coordenadas de este modelo ($\{\mathrm{MCS}\}$), se tiene que sus coordenadas son X_{MCS}, Y_{MCS} y Z_{MCS}. Con estas coordenadas y la función de proyección $g(x_{in}, y_{in})$ se obtiene la proyección en el plano ideal:

$$g(x_{in}, y_{in}) = \begin{bmatrix} x_{in} \\ y_{in} \\ f(\rho) \end{bmatrix} = \begin{bmatrix} \rho \cdot X_{MCS} / \sqrt{X_{MCS}^2 + Y_{MCS}^2} \\ \rho \cdot Y_{MCS} / \sqrt{X_{MCS}^2 + Y_{MCS}^2} \\ \rho \cdot Z_{MCS} / \sqrt{X_{MCS}^2 + Y_{MCS}^2} \end{bmatrix} \tag{5.5}$$

De la ecuación anterior tenemos la siguiente relación, que se utilizará para hallar el valor de ρ:

$$\frac{Z_{MCS}}{\sqrt{X_{MCS}^2 + Y_{MCS}^2}} \cdot \rho = a_4 \cdot \rho^4 + a_3 \cdot \rho^3 + a_2 \cdot \rho^2 + a_0 \tag{5.6}$$

donde a_4, a_3, a_2 y a_0 son los coeficientes del polinomio $f(\rho)$ que se obtuvieron previamente al calibrar con este modelo. Una vez conocida la distancia radial (ρ), las coordenadas en el plano ideal vienen dadas por:

$$\begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} = \begin{bmatrix} \rho \cdot X_{MCS} / \sqrt{X_{MCS}^2 + Y_{MCS}^2} \\ \rho \cdot Y_{MCS} / \sqrt{X_{MCS}^2 + Y_{MCS}^2} \end{bmatrix} \tag{5.7}$$

En este modelo las coordenadas ideales (x_{in}, x_{in}) y sus correspondientes en el plano imagen (u_{in}, v_{in}) están relacionadas por una matriz de transformación afín para compensar errores de desalineación y artefactos de digitalización, además de una traslación para que dichas coordenadas en píxel estén referidas a la esquina superior izquierda y no al centro. Por todo ello, la ubicación de la proyección en píxel (plano imagen) se obtiene mediante:

$$\begin{bmatrix} u_{in} \\ v_{in} \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} + \begin{bmatrix} c_u \\ c_v \end{bmatrix} \tag{5.8}$$

donde c, d y e son los elementos de la matriz afín y (c_u, c_v) el centro de la imagen. Tanto los elementos de la matriz afín como el centro se estiman durante el proceso de calibración junto con los parámetros del polinomio. Esta proyección se puede visualizar en la figura 5.4(b).

5.3.1.2.2. Proyección equidistante. Si se emplea una proyección *fisheye* como la equidistante no es necesario realizar un paso previo de calibración, sino conocer ciertas características físicas de las lentes como se verá a continuación.

Conociendo el punto 3D proyectado sobre la esfera y expresado en el sistema de referencia de la cámara, $\boldsymbol{P_{C_{del\|tras}}}$, el vector 3D que va desde el centro de la cámara a dicho punto, puede expresarse de la siguiente forma:

$$\vec{P}_{C_{del\|tras}} = \begin{bmatrix} x_{C_{del\|tras}} \\ y_{C_{del\|tras}} \\ z_{C_{del\|tras}} \end{bmatrix} = \begin{bmatrix} \sin\phi\cos\theta \\ \sin\phi\sin\theta \\ \cos\phi \end{bmatrix} \tag{5.9}$$

donde θ es el ángulo desde el eje $\mathrm{X}_{\mathrm{C_{del\|tras}}}$ positivo a la proyección del vector 3D $\vec{P}_{C_{del\|tras}}$ en el plano $\mathrm{X}_{\mathrm{C_{del\|tras}}}$-$\mathrm{Y}_{\mathrm{C_{del\|tras}}}$, mientras que ϕ, también conocido como ángulo Zenith, es el ángulo desde la dirección de vista de la cámara, es decir, su eje $\mathrm{Z}_{\mathrm{C_{del\|tras}}}$, al vector de coordenadas 3D.

La proyección de este punto 3D, $\boldsymbol{P_{C_{del\|tras}}}$, sobre la imagen *fisheye* puede expresarse en coordenadas polares (r, θ), donde la coordenada angular θ es el mismo ángulo que aparece en la ecuación (5.9). De este modo, se puede obtener su valor de la siguiente forma:

$$\theta = \tan^{-1}(y_{C_{del\|tras}}/x_{C_{del\|tras}}) \tag{5.10}$$

Con respecto a la coordenada radial (r), es decir, la distancia desde el centro de la imagen *fisheye* normalizada al punto proyectado, la proyección *fisheye* equidistante establece una relación lineal entre esta coordenada y ϕ:

$$r = \frac{r_{max}}{\phi_{max}} \cdot \phi \tag{5.11}$$

donde, por un lado, ϕ_{max} es el máximo valor que puede tomar el ángulo Zenith que viene determinado por la mitad del campo de visión de la lente, en radianes ($\phi_{max} = FOV_{rad}/2$). Por otro lado, la distancia radial máxima (r_{max}) es igual a uno pues las coordenadas de la imagen *fisheye* están normalizadas.

Después, se pasan las coordenadas polares (r, θ) a cartesianas (x_{in}, y_{in}):

$$(x_{in}, y_{in}) = (r \cdot \cos\theta, r \cdot \sin\theta) \tag{5.12}$$

Para finalizar, teniendo en cuenta que son coordenadas normalizadas (x_{in}, y_{in}), se realiza el proceso inverso para obtener sus coordenadas de píxel en la imagen *fisheye* original (u_{in}, v_{in}):

$$\begin{bmatrix} u_{in} \\ v_{in} \end{bmatrix} = \begin{bmatrix} a_x & 0 \\ 0 & a_y \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{5.13}$$

5.3.2 Proceso de registro de imágenes

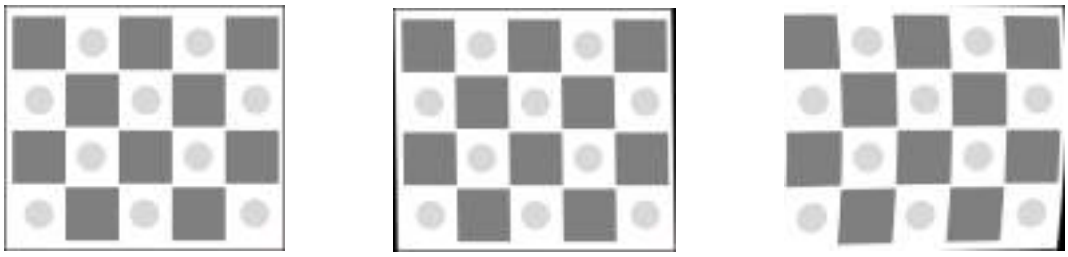
Cuando se dispone de dos o más imágenes tomadas desde distintos puntos de vista y el objetivo es combinarlas en una sola, todas estas imágenes deben expresarse en

un mismo sistema de referencia, de forma que la información común tenga la misma posición en la imagen final, en nuestro caso la vista esférica completa.

Con este propósito, en este capítulo se aplica un algoritmo de registro basado en características para alinear el par de imágenes equirectangulares. Así pues, se buscarán puntos característicos que son correspondencias entre ambas imágenes y estos se emplearán para estimar la transformación geométrica que las relaciona.

De este modo, esta parte del algoritmo se divide principalmente en los siguientes pasos. En primer lugar, dado el par de imágenes equirectangulares de entrada, se extraen los puntos característicos y sus respectivos descriptores. Después de esto, se realiza una búsqueda de correspondencias entre los dos conjuntos de puntos. Luego, con los pares de correspondencias obtenidos en el paso anterior, se estima la transformación geométrica. Para este paso existen dos posibilidades: utilizar la matriz afín 2D (apartado 5.3.2.1) o una transformación polinómica (apartado 5.3.2.2). Por último, la trasformación estimada se le aplicará a una de las imágenes, siendo esta la salida de esta etapa. Todos estos pasos se pueden visualizar gráficamente en la figura 5.6(a). En la figura 5.5, se muestra el resultado de aplicar a la imagen original (figura 5.5(a)) una transformación basada en una matriz afín (figura 5.5(b)) y basada en un polinomio (figura 5.5(c)).

Como se ha mencionado, en este capítulo, se utilizarán dos tipos de transformación geométrica. Por un lado, se encuentra la matriz afín que ya se ha utilizado en otros trabajos, como en [150], para este tipo de imágenes. Por otro lado, durante uno de los experimentos que realizamos para este capítulo (apartado 5.5.1.1) constatamos que la diferencia entre pares de puntos de vistas esféricas sigue un comportamiento no lineal y, como era de esperar, este comportamiento se acentúa en las partes más distorsionadas de la vista. Como consecuencia de esto, hemos propuesto utilizar una transformación basada en un polinomio para alinear el par de vistas esféricas.



(a) Original (b) Matriz afín (c) Polinomio

Figura 5.5: Transformaciones geométricas 2D. En (a) se muestra la imagen original, mientras que en (b) la resultante de aplicar una matriz afín y en (c) un polinomio de grado dos.

5.3.2.1 Matriz afín 2D

La matriz afín es una transformación geométrica que se caracteriza por preservar los puntos, las rectas y los planos, pero no los ángulos ni las longitudes (aunque sí sus relaciones) [350]. Esta transformación puede combinar traslaciones y transformaciones lineales, entre las que se encuentran la rotación, la escala, el cizallamiento y la reflexión.

Dado un punto (u, v), la transformación hacia delante para obtener el punto transformado (u', v') viene dada por la siguiente ecuación:

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{5.14}$$

siendo a, b, c y d los parámetros de la matriz referidos a las transformaciones lineales y los otros dos, t_x y t_y, a la traslación 2D.

5.3.2.2 Transformación polinómica

Como ya se ha comentado, en este trabajo proponemos aplicar una transformación geométrica basada en un polinomio debido al comportamiento de este tipo de imágenes. Esto se verá más detenidamente en apartado 5.5.1.1. De este modo, para alinear el par de vistas esféricas se estimará un polinomio de segundo grado que relacione los pares de correspondencias de puntos entre ellas.

La transformación polinómica 2D inversa viene dada por:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \end{bmatrix} \cdot \begin{bmatrix} v'^2 \\ u'^2 \\ u' \cdot v' \\ v' \\ u' \\ 1 \end{bmatrix} \tag{5.15}$$

siendo a_5, a_4, a_3, a_2, a_1, a_0 los coeficientes del polinomio para estimar la primera coordenada (u) y b_5, b_4, b_3, b_2, b_1, b_0 los coeficientes del polinomio para estimar la segunda coordenada (v).

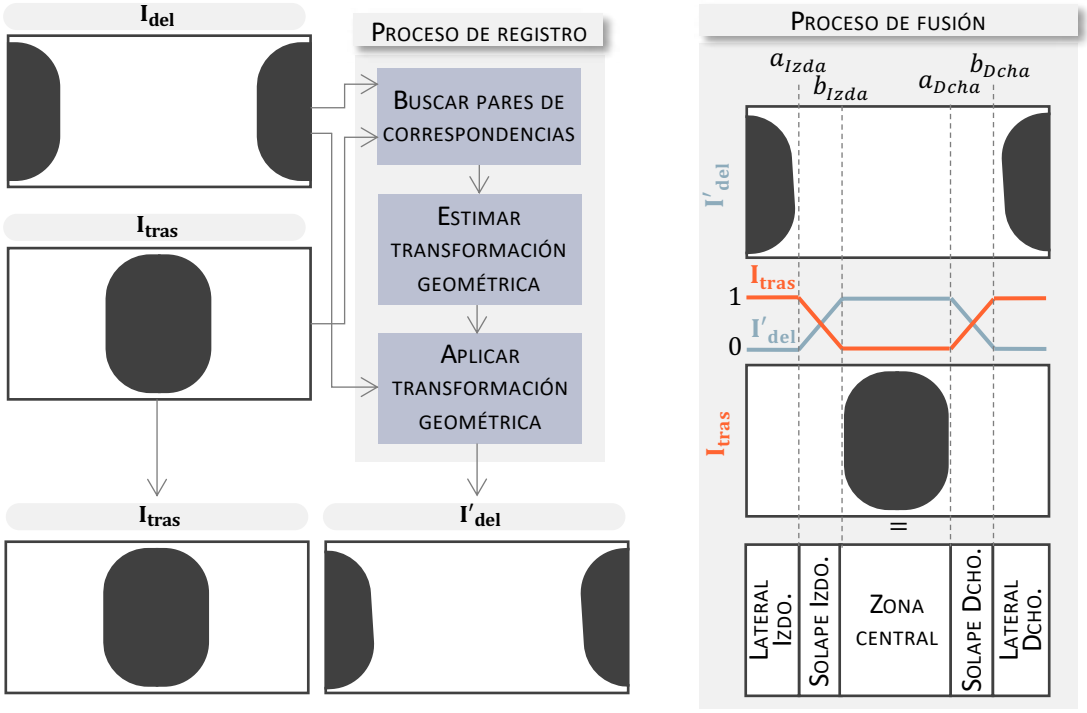
5.3.3 Proceso de fusión de imágenes

El último módulo de este algoritmo consiste en unir el par de imágenes obtenido anteriormente ($\mathbf{I'_{del}}$ y $\mathbf{I_{tras}}$) en una única imagen que sería la vista completa. Este módulo también es importante pues debe implementarse de forma que reduzca al mínimo las diferencias, en términos de apariencia global, causadas por posibles desajustes geométricos y/o fotométricos.

Los principales enfoques para realizar el proceso de fusión de imágenes son, por un lado, buscar el corte óptimo, y, por otro lado, el suavizado de la zona de transición (solape) entre las imágenes. Se puede encontrar más información acerca de estos enfoques en las referencias [351, 352].

En lo que respecta al primer enfoque, la finalidad es hallar la línea, en la zona de solape, en la que el corte es óptimo, es decir, la diferencia a ambos lados de esa línea es mínima. Este tipo de enfoque resulta interesante cuando hay presencia de artefactos debidos al paralaje u objetos en movimiento (escenas dinámicas). Esto se produce porque tienen en cuenta la información visual de la escena que aparece en la zona de

solape, por lo que no es una buena solución cuando existen diferencias de exposición o variaciones de iluminación de la escena entre las imágenes.



(a) (b)

Figura 5.6: Los dos últimos módulos del algoritmo: (a) proceso de registro y (b) proceso de fusión.

En cuanto al segundo enfoque, se realiza un suavizado de la información visual que aparece en la zona de solape. Así, se convierte en una solución apropiada ante discontinuidades causadas por desalineaciones fotométricas no debidas a errores durante el proceso de registro o presencia de objetos en movimiento. En este trabajo, hemos utilizado un método que pertenece a este segundo enfoque. Se trata de la función rampa empleada en [150].

En esta parte del algoritmo, se define la vista completa como un conjunto de 5 zonas (ver figura 5.6(b)): lateral izquierdo, solape izquierdo ($\mathbf{S_{Izda}}$), central, solape derecha ($\mathbf{S_{Dcha}}$) y lateral derecho. La función rampa únicamente se emplearán en las zonas de solapes.

Respecto a la función rampa, el valor de un píxel, con coordenadas (u, v), situado en una de las zonas de solape de la vista esférica completa vendrá dado por la siguiente ecuación:

$$\mathbf{S_i}(u,v) = \frac{v}{w} \cdot \mathbf{I}_{\mathbf{b_i^+}}(u,v) + \frac{w-v+1}{w} \cdot \mathbf{I}_{\mathbf{a_i^-}}(u,v) \tag{5.16}$$

donde i corresponde a la zona de solape ($Izda$ o $Dcha$) y w al ancho de la zona de solape. En cuanto a $\mathbf{I}_{\mathbf{a_i^-}}$ y $\mathbf{I}_{\mathbf{b_i^+}}$ corresponden a las zonas de solape del par de imágenes equirectangulares y dependen de cuál se esté procesando. Cuando se trata de la zona de solape izquierda ($\mathbf{S_{Izda}}$), por la izquierda tenemos la imagen equirectangular

trasera por lo que $\mathbf{I}_{\mathbf{a}_\mathbf{i}^-}$ es la zona de solape izquierda de esta imagen equirectangular, $\mathbf{I}_{\mathbf{a}_\mathbf{i}^-} = \mathbf{I}_{\mathbf{tras}}(:, a_{Izda} : b_{Izda}, :)$, mientras que por la derecha tenemos la imagen equirectangular delantera, de modo que $\mathbf{I}_{\mathbf{b}_\mathbf{i}^+}$ es la zona de solape izquierda de esta imagen equirectangular, $\mathbf{I}_{\mathbf{b}_\mathbf{i}^+} = \mathbf{I}_{\mathbf{del}}(:, a_{Izda} : b_{Izda}, :)$. En el caso de la zona de solape derecha ocurre lo contrario, $\mathbf{I}_{\mathbf{a}_\mathbf{i}^-}$ es la zona de solape derecha de la imagen equirectangular delantera, $\mathbf{I}_{\mathbf{a}_\mathbf{i}^-} = \mathbf{I}_{\mathbf{del}}(:, a_{Dcha} : b_{Dcha}, :)$, a diferencia de $\mathbf{I}_{\mathbf{b}_\mathbf{i}^+}$ que es es la zona de solape derecha de la imagen equirectangular trasera, $\mathbf{I}_{\mathbf{b}_\mathbf{i}^+} = \mathbf{I}_{\mathbf{tras}}(:, a_{Dcha} : b_{Dcha}, :)$.

5.4 Enfoques para evaluar la calidad de la zona de solape

El objetivo de este capítulo es conseguir una vista esférica completa en la que los artefactos y las discontinuidades producidas al fusionar la información visual en las zonas de solape sean lo más reducidas posible, teniendo así una imagen con una calidad alta. Así pues, si observamos la vista esférica completa que se ha generado podemos decir si tiene una buena calidad o no, de forma cualitativa (como se hará en el apartado 5.7.2.5), observando si aparecen ciertos efectos producidos por una combinación (alineación y fusión) incorrecta. Sin embargo, este capítulo también tiene como objetivo realizar una evaluación cuantitativa.

Dado que estos efectos aparecen en las zonas en las que se fusionan la información capturada por las dos imágenes (es decir, en la zona de solape), los métodos de evaluación se aplicarán a dichas zonas. Como se tienen dos zonas de solape, izquierda y derecha, se evaluará cada una de ellas por separado. A continuación, se describen los tres métodos utilizados en este capítulo para evaluar las vistas esféricas completas.

5.4.1 Medida de la calidad sin referencia

Cuando se combina la información común de dos imágenes, si la alineación entre ellas no se ha estimado correctamente puede que aquellas zonas ricas en textura no se vean con nitidez. De este modo, se describe una medida para evaluar la calidad de la imagen (IQ, siglas del inglés *Image Quality*) en función de la nitidez y sin necesidad de una imagen de referencia.

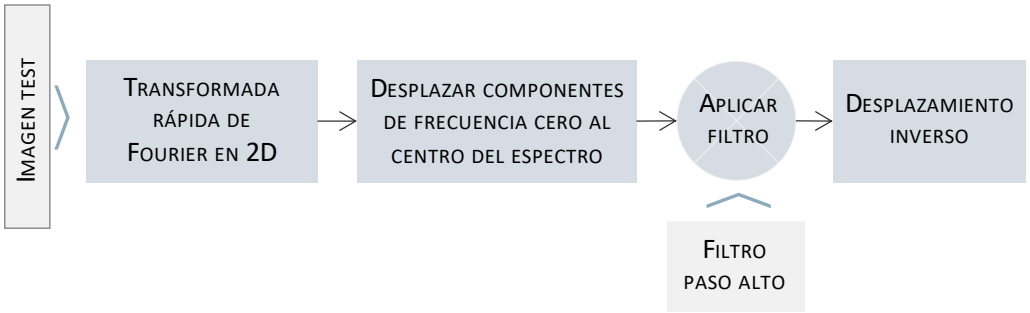


Figura 5.7: Diagrama de bloques para la obtención de la medida de calidad basada en la nitidez como resultado de realizar un estudio en el dominio de la frecuencia.

El procedimiento para calcular esta medida, como puede verse en la figura 5.7, comienza por obtener la transformada de Fourier 2D discreta de la región a evaluar.

Para ello se emplea un algoritmo de transformada rápida de Fourier. El siguiente paso consiste en aplicar un filtro de paso alto que elimina las frecuencias bajas, pero antes de esto se deben desplazar los componentes de frecuencia cero al centro. Tras aplicar este filtro, se vuelven a desplazar los componentes de frecuencia cero a su ubicación original ($\mathcal{F}_\mathbf{i}$) y se calcula la media del espectro de magnitud tal y como se define con la siguiente ecuación:

$$IQ_{nitidez} = \frac{\sum_{i=1}^{M}(1 + |\mathcal{F}_i(u,v)|)}{M} \tag{5.17}$$

donde M es el número total de píxeles en la región solapada. La imagen evaluada tendrá mayor nitidez cuanto mayor sea el valor de esta medida. La desventaja que presenta esta medida es que dependerá de la cantidad, forma y gradiente de los bordes presentes en la zona de solape. Así pues, no puede tomarse como una medida de calidad absoluta.

5.4.2 Medida de la calidad con referencia

En la literatura podemos encontrar una variedad de enfoques que establecen una medida de cómo de buena es una imagen tras compararla con otra de referencia. No obstante, muchos de ellos no ofrecen unos resultados de calidad que se asemejen a los que se llegaría mediante la percepción del ser humano, pues esta se acerca más a la extracción de la información estructural de la escena. En esto último se basa el método de indexación por similitud estructurada (SSIM, acrónico del inglés *Structured Similarity Indexing Method*) [13].

La puntuación de calidad de imagen que proporciona SSIM es el resultado de comparar ambas imágenes de entrada (la de test y la de referencia) bajo tres aspectos como son la luminancia, el contraste y la estructura. Finalmente, la combinación de las funciones para cada uno de estos aspectos se establece como puntuación. Sin embargo, es un método que utiliza una escala única, lo que puede conllevar a que no sea apropiado para todos los entornos. Con respecto a esto, aparece una versión ampliada de este procedimiento que se conoce como índice de similitud estructural multiescala (MS-SSIM, acrónimo del inglés *Multi Scale Structural Similarity Index Method*) [193] y que lleva la evaluación a distintas escalas de imagen. Para esto último, se aplican a las entradas del algoritmo, $\mathbf{I_{test}}$ e $\mathbf{I_{ref}}$, de forma iterativa, un filtro paso bajo y un posterior muestreo descendente de la imagen por un factor de dos, hasta un total de $M-1$ veces. Dentro de estas múltiples escalas, la primera corresponde a la imagen de resolución original y la escala M a la resolución más baja.

MS-SSIM se caracteriza por realizar, en varias escalas (j), tanto la comparación de información estructural (s_j) como la comparación en cuanto al contraste (c_j), a diferencia de la comparación de luminancia que únicamente se lleva a cabo en la escala más alta (M). De modo que, la puntuación de calidad de imagen proporcionada por MS-SSIM se calcula como:

$$IQ_{MS-SSIM}\ (\mathbf{I_{test}}, \mathbf{I_{ref}}) = \\ l_M(\mathbf{I_{test}}, \mathbf{I_{ref}})]^{\alpha_M} \prod_{j=1}^{M} [c_j(\mathbf{I_{test}}, \mathbf{I_{ref}}]^{\beta_j} [s_j(\mathbf{I_{test}}, I_{red})]^{\gamma_j}) \tag{5.18}$$

donde cada uno de los exponentes (α_M, β_j y γ_j) se emplean para ajustar la importancia relativa de cada uno de los términos.

5.4.3 Evaluación basada en *deep learning*

En este capítulo, como se ha comentado al inicio, hemos propuesto un método de evaluación cuya medida de calidad se obtiene como resultado de calcular la similitud entre el descriptor de apariencia global de la zona de solape a evaluar y el descriptor de la de referencia, donde ambos descriptores se han generado a partir de una arquitectura de red neuronal.

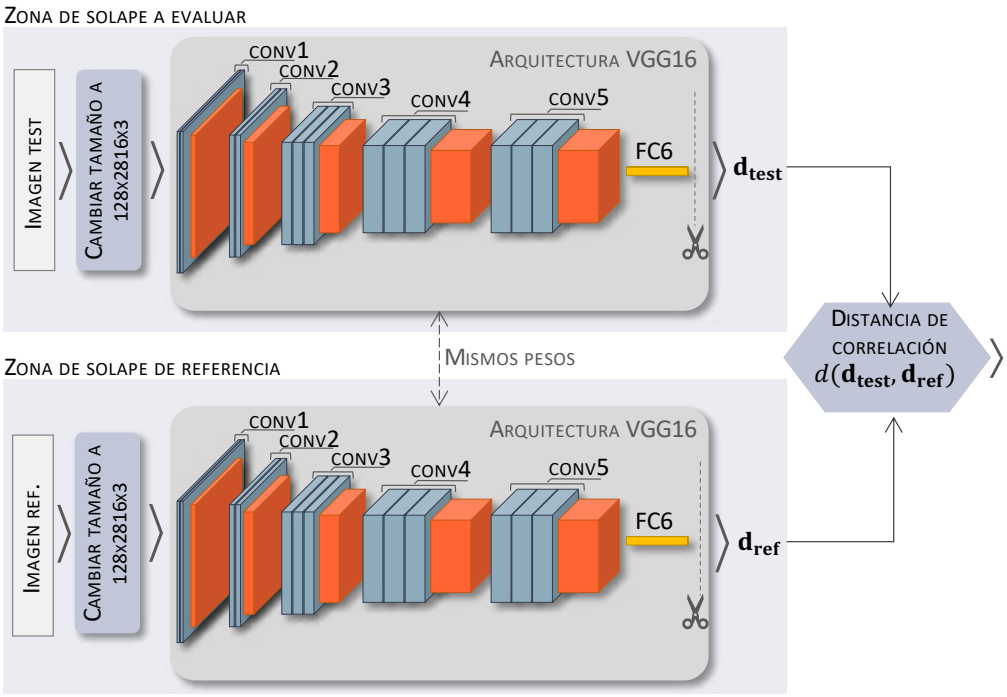


Figura 5.8: Diagrama de bloques para la obtención de la medida de calidad basada en la distancia entre dos descriptores de apariencia global obtenidos a partir de dos redes neuronales con misma arquitectura y pesos.

Según se muestra en la figura 5.8, este algoritmo requiere de dos entradas. Por un lado, una de las entradas es la región de la vista esférica completa que se quiere evaluar, por lo que corresponde a la zona en la que se ha fusionado la información capturada por ambas cámaras (zona de solape). Por cada vista esférica completa se tiene dos zonas de solape, pero como se ha mencionado anteriormente, se evaluarán de forma independiente, de manera que esta región de entrada será una de ellas. Por otro lado, la otra entrada será una región, denominada de referencia, que tiene el mismo tamaño que la anterior y en la que aparece también la misma información del entorno, pero sin ningún efecto debido a la fusión. En otras palabras, esta segunda región es el *ground truth*, que define cómo sería la imagen si la combinación de la información común a ambas cámaras es correcta.

Atendiendo a la figura 5.8, este algoritmo de evaluación se compone de dos etapas, una primera que consiste en extraer un descriptor para cada una de las regiones

descritas y una segunda etapa en la que se calcula la distancia de correlación entre ambos descriptores, teniendo así una medida de diferencia entre las regiones de entrada. Cuanto menor sea esta medida menor será la diferencia, y como consecuencia, mayor será la calidad de la zona de solape evaluada.

La etapa de extracción de los descriptores se compone de dos arquitecturas de red que comparten pesos y que además se componen de las mismas capas. Ambas arquitecturas corresponden a una parte de la VGG16 [353].

La red VGG16 se compone de 13 capas convolucionales, 5 capas *max-pooling* y 3 capas *fully-connected*. Además, esta red ha sido preentrenada para clasificar diferentes categorías de objetos en imágenes. El objetivo no es resolver un problema de clasificación, sino obtener un descriptor que contenga las características más relevantes de esta región de imagen. De esta forma, en el algoritmo solo se implementa hasta la primera capa *fully-connected* de la arquitectura de esta red (como puede verse en la figura 5.8). Entre todas las capas se ha seleccionado esta por los estudios desarrollados en trabajos anteriores [329, 354]. Asimismo, el descriptor será la salida de esta capa *fully-connected* por lo que tendrá un tamaño de 1x4096.

Tal y como muestra la figura 5.8, lo comentado en el párrafo anterior se lleva a cabo tanto para la zona de solape que se pretende evaluar como para la zona de solape de referencia, la cual es una región del mismo tamaño y con la misma información del entorno pero que no sufre artefactos debido a la combinación de ambas imágenes. Cabe indicar que las regiones de entrada al algoritmo de evaluación son escaladas a un tamaño de 128x2816x3 antes de aplicarse la red neuronal. Finalmente, se obtienen dos descriptores holísticos ($\mathbf{d_{test}}$ y $\mathbf{d_{ref}}$) con los que se calcula la distancia de correlación, $d(\mathbf{d_{test}}, \mathbf{d_{ref}})$, entre ambos descriptores, consiguiendo así la medida de calidad de la zona de solape. Esta medida, dado que se basa en una distancia, mide la diferencia entre la zona a evaluar y la de referencia que representa el mejor caso, así pues, cuanto menor sea esta medida mejor calidad relativa presentará esa zona de solape de la imagen de test que se está evaluando.

5.5 Corrección entre el par de imágenes *fisheye*

Si el par de imágenes *fisheye* fuesen tomadas por la misma cámara con una orientación relativa pura de $180°$ respecto al eje vertical, los pares de correspondencias tendrían la misma ubicación en el plano imagen tras aplicar dicha rotación. Sin embargo, esto no ocurre debido a que no es el mismo dispositivo el que captura ambas imágenes, además de que existe, aunque pequeña, una traslación relativa entre ellas (los centros ópticos no son coincidentes) y puede que la orientación entre ellas no sea una transformación pura como se ha dicho.

5.5.1 Estudio de la relación entre pares de correspondencias

En vista del párrafo anterior, el objetivo de este apartado es estudiar esta diferencia de posición entre pares de correspondencias de puntos. Por un lado, se analizarán los pares de imágenes equirectangulares (antes de aplicar la transformación geométrica). Por otro lado, también se estudiará esta diferencia en los pares de imágenes *fisheye*. Como se comentará cuando se describa este estudio, esto último se ha realizado con

el objetivo de corregir esta diferencia entre pares de correspondencias encontradas en este tipo de imágenes.

5.5.1.1 Estudio de imágenes equirectangulares (coordenadas cartesianas)

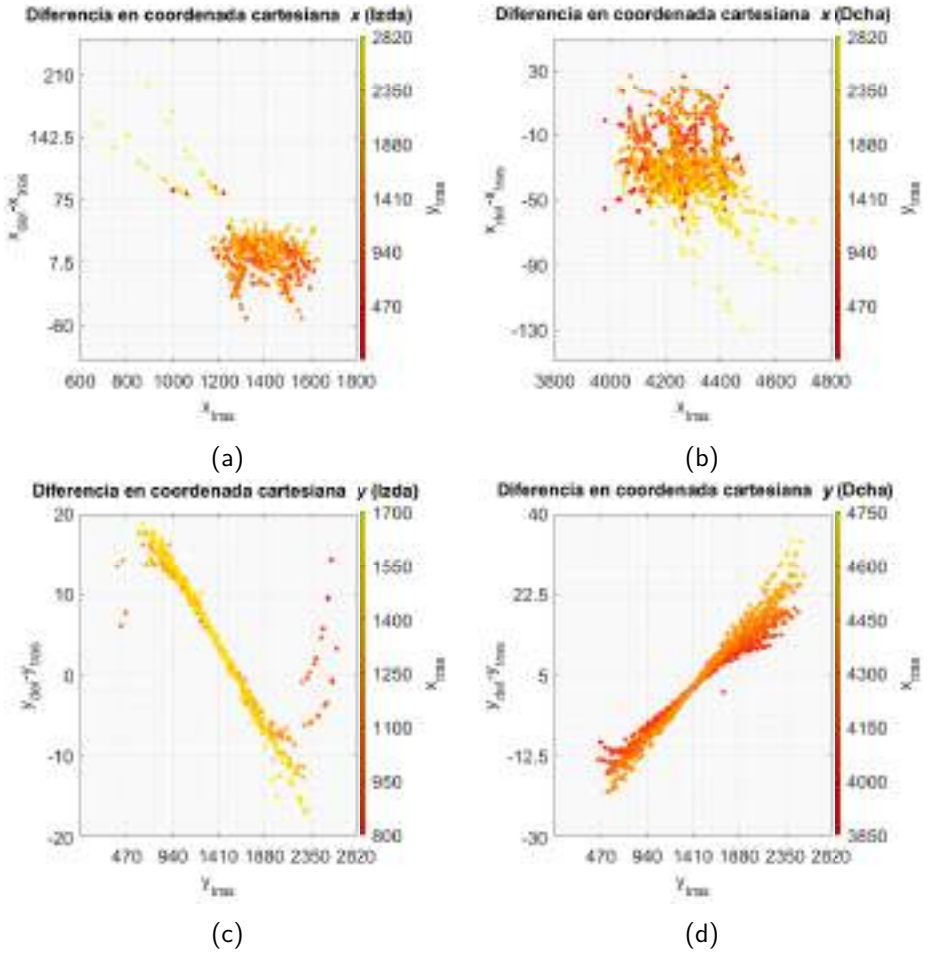
El primer estudio se lleva a cabo con imágenes en formato equirectangular pues es el tipo de imágenes con el que se estima y aplica la transformación geométrica 2D para alinear ambas imágenes. Asimismo, como este estudio se basa en pares de correspondencias, estas se han obtenido a partir de las esquinas detectadas de marcas ArUco, como se explica a continuación.

El procedimiento para este primer estudio se divide en un primer paso que es la adquisición de las imágenes. Para dicho fin, en primer lugar, la cámara se dispuso de forma que capturase, en la zona de solape, marcas ArUco con diferentes identificadores. Tras adquirir el primer par de imágenes *fisheye* en esta posición, se aplicaron varias rotaciones a la Garmin VIRB 360 sobre el eje óptico con el propósito de que las marcas ArUco cubrieran toda la zona de solape. El siguiente paso fue transformar este conjunto de pares de imágenes *fisheye* a formato equirectangular, tal y como se expone en el apartado 5.3.1. Después, se utilizó un método de identificación de marcas ArUco así como de detección de las esquinas de estas marcas. De modo que, para cada marca identificada en una de las imágenes, se tienen cuatro puntos que corresponden a sus esquinas. Para finalizar, en cada par de imágenes equirectangulares, se comprueba qué marcas han sido identificadas en ambas imágenes, teniendo así cuatro pares de correspondencias por marca. Tras realizar este proceso con todos los pares de imágenes, se hallaron un total de 1628 pares de correspondencias. Cabe señalar que el estudio se hará de forma independiente en cada zona de solape.

En primer lugar, se analiza la diferencia entre las coordenadas cartesianas de los pares de correspondencias encontradas, cuyos resultados se visualizan en la figura 5.9. Por un lado, se calcula la diferencia entre las coordenadas cartesianas x. En este sentido, la figura 5.9(a) muestra los valores de la zona de solape izquierda, mientras que la figura 5.9(b) muestra los de la zona de solape derecha. Por otro lado, también se estudia la diferencia entre las coordenadas cartesianas y, correspondiendo la figura 5.9(c) a la zona de solape izquierda y la figura 5.9(d) a la derecha. En las cuatro gráficas, los valores obtenidos al calcular la diferencia entre las coordenadas cartesianas ($x_{del} - x_{tras}$ o $y_{del} - y_{tras}$) se representan en el eje vertical, mientras que el eje horizontal de la gráfica corresponde a la coordenada de la correspondencia en la imagen equirectangular trasera (x_{tras} o y_{tras}, respectivamente). Asimismo, el color asignado representa el valor de la otra coordenada cartesiana (y_{tras} o x_{tras}).

Analizando los valores obtenidos como diferencia de las coordenadas horizontales (x) vemos que esta es mayor cuando los puntos de las correspondencias se encuentran en la zona superior (valores de la coordenada y más altos) y en la zona inferior (valores de la coordenada y más bajos). También observamos que la diferencia es más notable cuanto más alejados se encuentre del centro horizontal ($x = 1410/x = 4230$) de la zona de solape izquierda/derecha, respectivamente. Al estudiar los valores obtenidos como diferencia de las coordenadas verticales (y), notamos una relación más evidente. Mientras que esta diferencia se puede definir como prácticamente lineal cuando las correspondencias se encuentran en la zona central ($y = 1610$), a medida que la coor-

denada cartesiana y adquiere un valor mayor o menor esta linealidad va disminuyendo. En otras palabras, en las zonas superior e inferior de la imagen equirectangular (los polos de la esfera) la diferencia entre las coordenadas cartesianas y es no lineal.



(a) (b) (c) (d)

Figura 5.9: Estudio de la diferencia entre las coordenadas cartesianas de los pares de correspondencias encontrados en los pares de imágenes equirectangulares. Diferencia entre las coordenadas x en la zona de solape (a) izquierda y (b) derecha. Diferencia entre las coordenadas y en la zona de solape (c) izquierda y (d) derecha.

En segundo lugar, se analiza la relación entre las coordenadas cartesianas de los pares de correspondencias encontradas, cuyos resultados se visualizan en la figura 5.10. Por un lado, se representa la relación entre las coordenadas cartesianas x en la zona de solape izquierda en la figura 5.10(a) y la zona de solape derecha en la figura 5.10(b). Por otro lado, la relación entre las coordenadas cartesianas y se puede observar en la figura 5.10(c) para la zona de solape izquierda y en la figura 5.10(d) para la zona de solape derecha. En las cuatro gráficas, se representa la coordenada cartesiana de la imagen equirectangular delantera (x_{del} o y_{del}) en el eje vertical frente a la coordenada cartesiana de la imagen equirectangular trasera (x_{tras} o y_{tras}) en el eje horizontal. En este caso, el color define el valor de diferencia de las coordenadas cartesianas que se esté representando ($x_{del} - x_{tras}$ o $y_{del} - y_{tras}$).

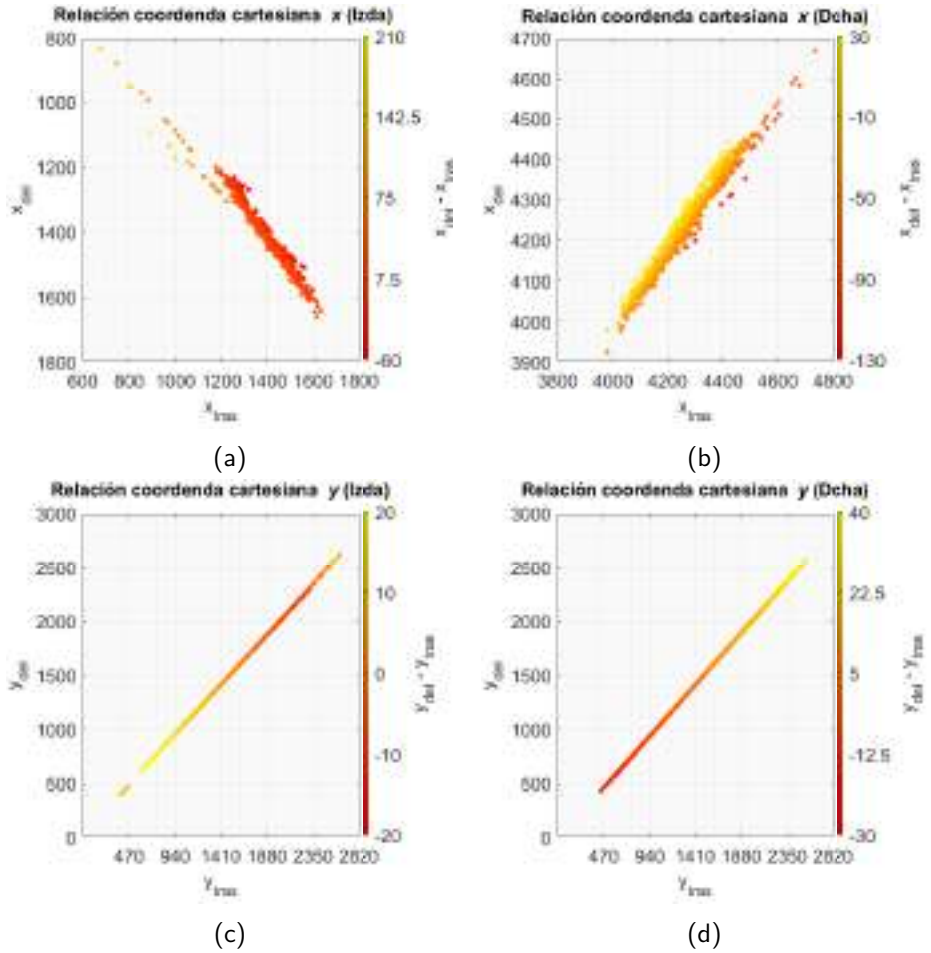


Figura 5.10: Estudio de la relación entre las coordenadas cartesianas de los pares de correspondencias encontrados en los pares de imágenes equirectangulares. Relación entre las coordenadas x en la zona de solape (a) izquierda y (b) derecha. Relación entre las coordenadas y en la zona de solape (c) izquierda y (d) derecha.

Una vez observado los resultados mostrados en estas gráficas, en el caso de la coordenada cartesiana x no se observa una tendencia clara, por el contrario, esto sí ocurre con la coordenada cartesiana y, donde se observa una relación prácticamente lineal. En los resultados de la zona de solape izquierda (figura 5.10(c)), se observa que la diferencia entre estas coordenadas cartesianas es positiva y máxima para valores bajos de y (parte superior de la imagen equirectangular) a medida que aumenta el valor de y esta diferencia se hace menor hasta llegar a conseguir un valor mínimo y negativo para valores altos de esta coordenada (parte inferior de la imagen equirectangular). Por el contrario, en la zona de solape derecha se observa este comportamiento, pero inverso, es decir, la diferencia entre las coordenadas y es mínima y negativa para valores bajos de esta coordenada y se hace mayor y positiva a medida que la coordenada y adquiere un valor mayor.

De este estudio de la relación entre los pares de imágenes equirectangulares, destacamos lo siguiente:

- La relación entre las correspondencias en formato equirectangular presenta una

no linealidad, la cual se acentúa en la parte superior e inferior de este tipo de imágenes.

Así concluimos que una transformación lineal (como sería la matriz afín) producirá buenos resultados en la zona central que es donde se cumple cierta linealidad, pero no en la parte superior e inferior. Atendiendo a esto, proponemos aplicar una transformación geométrica no lineal, como sería un polinomio de grado dos (apartado 5.3.2.2), para alinear el par de imágenes equirectangulares (proceso de registro de imágenes).

5.5.1.2 Estudio de imágenes *fisheye* (coordenadas polares)

Como se ha mencionado, el proceso de registro entre imágenes se suele realizar con los pares de imágenes equirectangulares. El principal propósito es minimizar la diferencia estudiada en el apartado 5.5.1.1 tras aplicar la transformación estimada. Sin embargo, uno de los objetivos de este capítulo es minimizar esta diferencia antes.

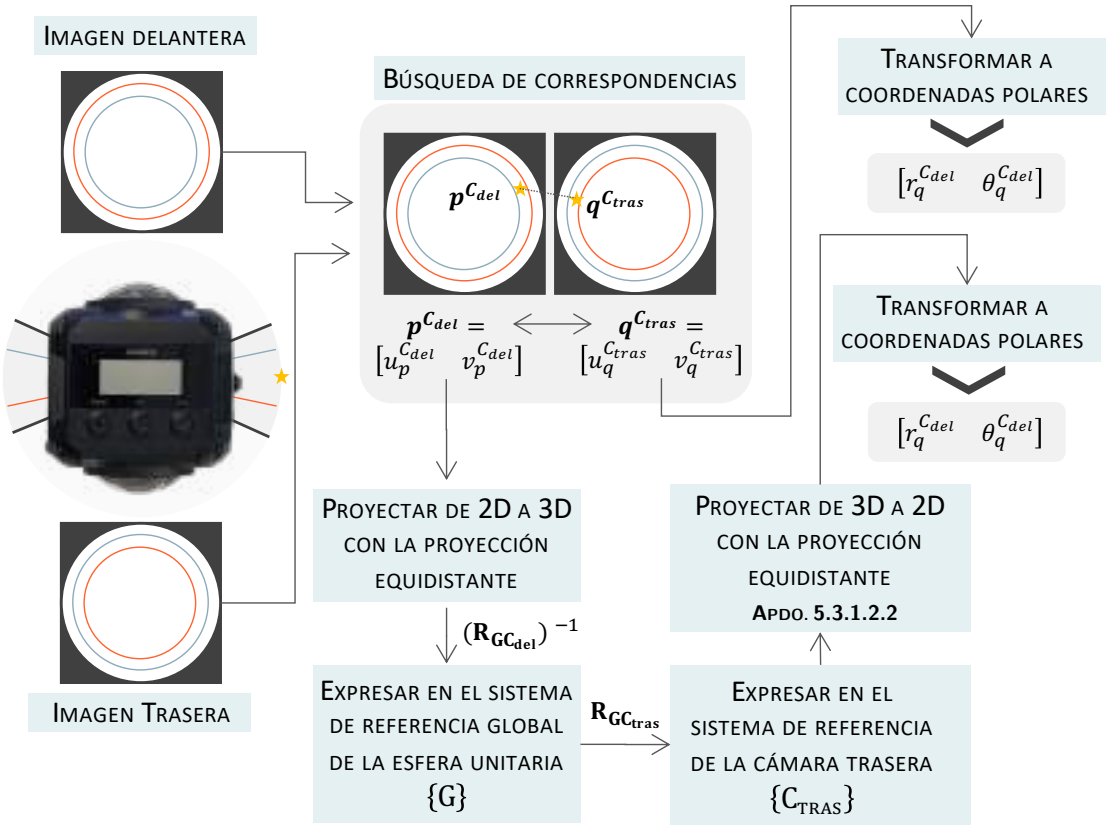


Figura 5.11: Diagrama de bloques con los pasos principales para expresar los pares de correspondencias entre el par de imágenes *fisheye* en el mismo plano imagen y en formato polar.

Asumiendo un caso ideal, en el que el par de imágenes *fisheye* ha sido capturado por el mismo dispositivo y con una rotación pura de $180°$, si se tiene un par de correspondencias de estas imágenes ($\boldsymbol{p}^{C_{del}} \longleftrightarrow \boldsymbol{q}^{C_{tras}}$), lo que se esperaría es que al proyectar el punto detectado en una de ellas ($\boldsymbol{p}^{C_{del}}$) al espacio 3D (esfera unitaria), aplicarle la rotación mencionada (nueva pose) y proyectarlo de nuevo al plano imagen *fisheye*, su

ubicación coincida con la del punto en la otra imagen ($\boldsymbol{q}^{C_{tras}}$). Sin embargo, esto no ocurre por lo que, en este estudio, se va a estudiar la diferencia entre correspondencias encontradas en pares de imágenes *fisheye*.

Para poder estudiar la relación entre las correspondencias del par de imágenes *fisheye*, se implementa un algoritmo con los pasos mencionados en el párrafo anterior. Esto se puede apreciar en la figura 5.11, donde la entrada es un par de imágenes *fisheye* y a la salida se tienen pares de correspondencias entre ellas pero las coordenadas de los puntos característicos de la imagen *fisheye* trasera ($tras$) se encuentran expresadas en el sistema de referencia de la imagen *fisheye* delantera (del). Se asume que no existe traslación entre los dos sistemas de coordenadas de las cámaras y que además la rotación relativa es la que se indica en la figura 5.4(a) (caso ideal).

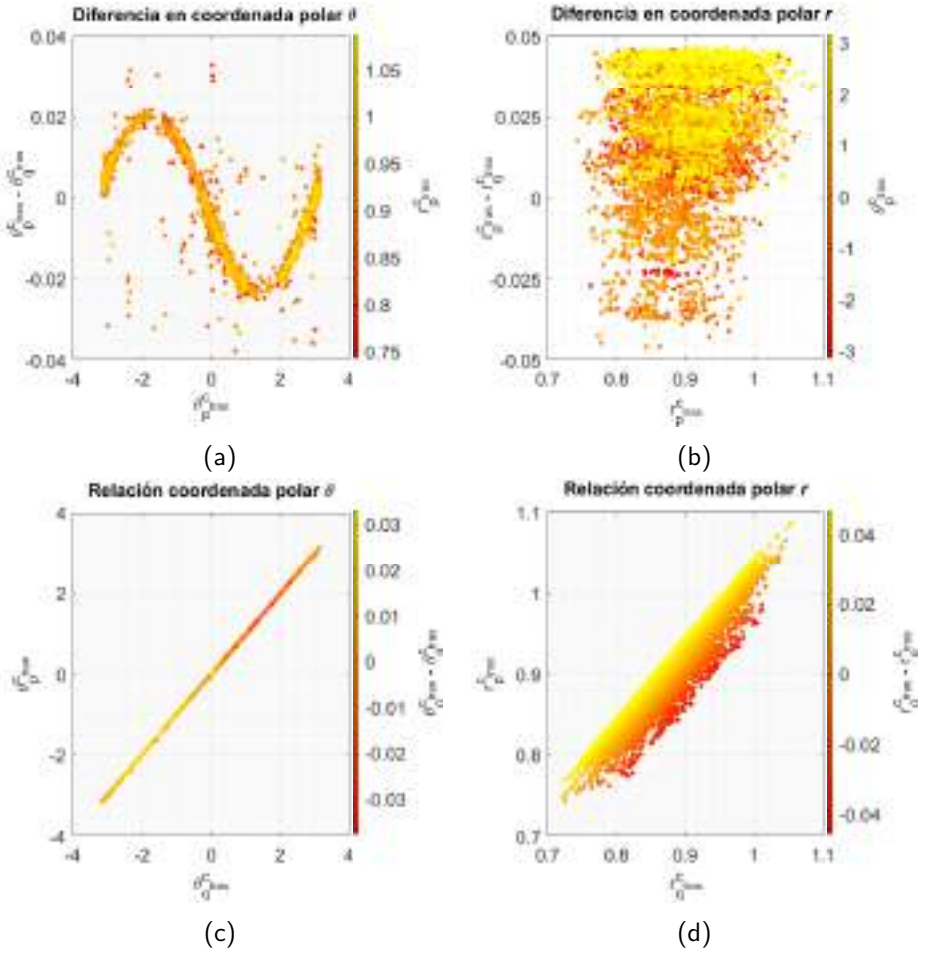


Figura 5.12: Estudio de las coordenadas polares. Diferencia entre las coordenadas polares (a) θ y (b) r. Relación entre las coordenadas polares (c) θ y (d) r.

De este segundo estudio sobre la relación entre los pares de imágenes *fisheye*, destacamos lo siguiente:

- La diferencia entre las coordenadas θ de los pares de correspondencias entre las imágenes *fisheye* en un mismo plano imagen ($\theta_p^{C_{tras}} - \theta_q^{C_{tras}}$) puede modelarse con una función seno que depende de la coordenada asociada a la imagen *fisheye*

delantera pero expresada en el plano imagen trasera ($\theta_p^{C_{tras}}$) .

$$\theta_p^{C_{tras}} - \theta_q^{C_{tras}} = a \cdot \sin\left(b \cdot \theta_p^{C_{tras}} + c\right) \tag{5.19}$$

- La relación entre distancias radiales de las coordenadas polares se puede decir que viene dada por un factor de escala α, que se puede estimar con la siguiente ecuación:

$$\alpha = \sum_{j=1}^{N} \frac{r_{q_j}^{C_{tras}}}{r_{p_j}^{C_{tras}}} \tag{5.20}$$

5.5.2 Implementación de la corrección en el algoritmo

Tras las dos conclusiones a las que se ha llegado con el estudio de la relación entre coordenadas polares del par de imágenes *fisheye* (apartado 5.5.1.2), hemos implementado pasos adicionales en el primer módulo del algoritmo con el objetivo de minimizar el desajuste entre correspondencias del par de imágenes considerando su relación en el plano imagen *fisheye*.

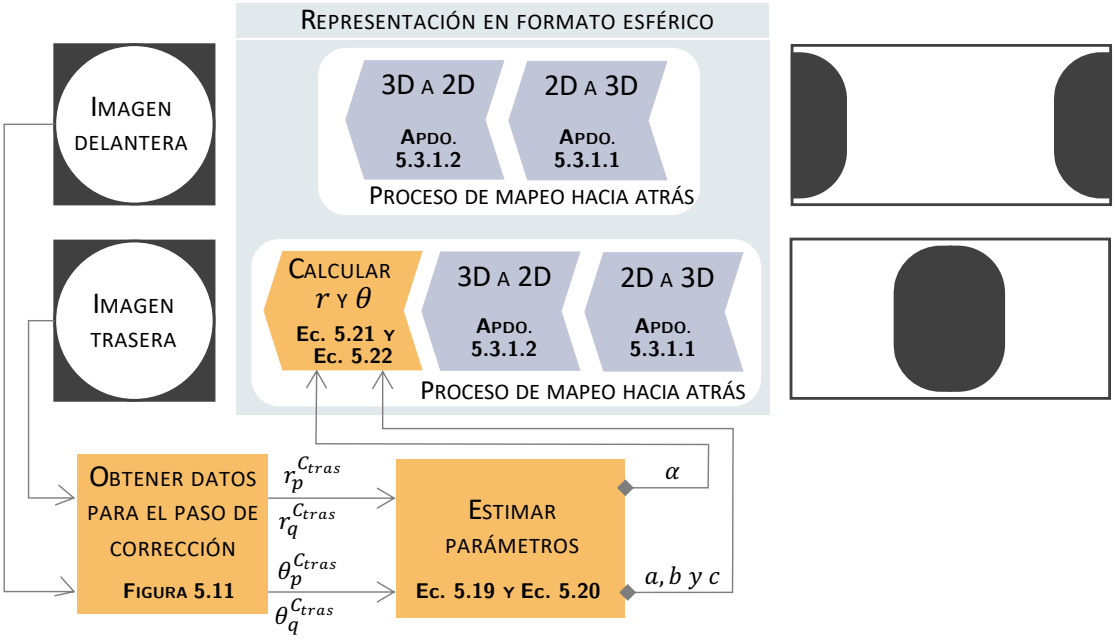


Figura 5.13: Módulo de representación en formato esférico modificado para implementar el paso de corrección.

De forma que, ahora el primer módulo del algoritmo es el que se muestra en la figura 5.13. Dado un par de imágenes *fisheye*, en primer lugar, se obtienen los datos necesarios para estimar los parámetros de dicha relación ($\theta_p^{C_{tras}}$, $r_p^{C_{tras}}$, $\theta_q^{C_{tras}}$ y $r_q^{C_{tras}}$), para ello se ejecuta el algoritmo de la figura 5.11. En segundo lugar, con estos datos, junto con las ecuaciones (5.19) y (5.20), se estiman los parámetros requeridos: a, b, c y α. En tercer lugar, estos parámetros se introducen en el paso de representación en formato esférico de la imagen *fisheye* trasera y se emplean para corregir θ' y ϕ'

que se han obtenido empleando la proyección equidistante del apartado 5.3.1.2.2, es decir, las ecuaciones (5.11) y (5.10). En otras palabras, los valores de θ' y ϕ' son los hipotéticos, es decir los que se tendría en un caso ideal donde la imagen *fisheye* trasera se ha obtenido rotando la misma cámara (sensor) $180°$ sobre su eje vertical (como en la figura 5.4(a)) y con traslación nula. Dado que esto no es así, como ya se ha comentado, se intenta obtener los reales empleando las dos relaciones descritas en el apartado anterior. Así pues, finalmente se obtienen las coordenadas polares ya corregidas (r y θ):

$$\theta = \theta' - \sin\left(b \cdot \theta' + c\right) \tag{5.21}$$

$$r = \alpha \cdot a \cdot \phi' \tag{5.22}$$

Para finalizar, aplicando las ecuaciones (5.13) y (5.12) con los valores obtenidos de r y θ, se tienen las coordenadas en el plano imagen *fisheye*.

5.6 *Dataset* de imágenes

Como se ha comentado a lo largo de este capítulo, la cámara escogida para llevar a cabo los objetivos es la Garmin VIRB 360. Así pues, con esta cámara se ha capturado un conjunto de imágenes en distintos formatos para poder llevar a cabo los experimentos del siguiente apartado.

Esta cámara se puede configurar con distintos modos (tabla 3.2) de forma que, para cada uno de ellos, proporciona un tipo distinto de imagen. Teniendo en cuenta que el objeto de estudio de este capítulo consiste en, a partir de un par de imágenes *fisheye*, generar vistas esféricas completas y, además, comprarlas con la que proporciona la cámara, durante la adquisición del *dataset* solo se selecciona el modo de *360* y el modo *RAW*.

Para poder realizar estudios lo más completos posibles, la adquisición del *dataset* se ha realizado en cuatro estancias diferentes del edificio Innova de la Universidad Miguel Hernández de Elche. Puesto que el algoritmo empleado se basa en puntos característicos y que, además, se ha observado que la vista esférica completa proporcionada por la Garmin tiene peor calidad cuando la zona de solape es rica en textura, se han seleccionado dichas estancias con el objetivo de tener imágenes con un amplio rango de cantidad en cuanto a información visual se refiere. En otras palabras, para un estudio más balanceado y que no favorezca a un tipo de método de generación de imagen (basado en características) u otro, el *dataset* empleado se compone de imágenes con mayor información visual (que se espera que beneficie a las generadas empleando el algoritmo descrito en este capítulo) e imágenes con menor información visual (que se espera que beneficie a la propia de la cámara Garmin). En la tabla 5.1, se describen estas cuatro estancias en cuanto a información visual presente; número de ubicaciones en las que se ha capturado un par de imágenes *fisheye* y una vista esférica completa con el modo *360*; y el área aproximada de esa estancia.

5.7 Experimentos y resultados

En el apartado 5.3, se ha descrito el algoritmo para generar vistas completas a partir de pares de imágenes *fisheye*. Además, a lo largo de este capítulo, se han propuesto

Tabla 5.1: Tabla con información de las estancias en las que se han capturado las imágenes del *dataset*.

	Descripción	Núm. de ubicaciones	Área
Oficina	Las imágenes capturadas en esta estancia presentan ciertos objetos característicos, como ordenadores, escritorios, armarios, percheros, y una pizarra blanca. Además, en las paredes hay colgados carteles y algunas marcas ArUco dispuestas en armarios o paredes de forma que aumentan la presencia de textura en las imágenes.	24 ubicaciones	$27m^2$
Laboratorio	Esta estancia presenta una gran cantidad y variedad de objetos, aportando más información visual. Pese a esto, hay que tener en cuenta que es la estancia más grande, por lo que los objetos suelen aparecer en la zona menos problemática de la vista esférica completa, es decir, en la zona central.	6 ubicaciones	$117m^2$
Sala de reuniones	Los objetos que podemos encontrar en estas imágenes son estanterías con libros, sillas y escritorios, entre otros no tan frecuentes. La principal característica de este escenario es que existe bastante simetría en cuanto a aspecto visual y que además la información visual es repetitiva.	11 ubicaciones	$55m^2$
Pasillo (planta 2)	Las imágenes capturadas en esta estancia son poco ricas en información visual ya que las estructuras predominantes son paredes, grandes ventanales y puertas, únicamente hay una mayor textura por las señales de emergencia o los carteles informativos.	9 ubicaciones	$69m^2$
Pasillo (planta 0)	Las características de esta estancia son parecidas a la anterior pero con menor información visual relevante pues solo hay señales de emergencia.	2 ubicaciones	$50m^2$

varias opciones a escoger en algunos de los pasos. En lo que respecta a la proyección desde la imagen *fisheye* a la esfera unitaria, este mapeo se puede realizar mediante la función de proyección equidistante (apartado 5.3.1.2.2) o mediante el modelo de cámara propuesto por Scaramuzza et al. [10] (apartado 5.3.1.2.1). En el primer experimento (apartado 5.7.1) de este apartado, se realiza una evaluación y comparación de las vistas generadas con estos dos procedimientos (junto con la que proporciona la cámara

Garmin VIRB 360).

Asimismo, en este capítulo se ha propuesto un paso de corrección para minimizar la diferencia de coordenadas entre el par de imágenes equirectangulares antes de la transformación a dicho formato. Como se ha comentado en el apartado 5.5.2, este paso de corrección se realiza durante el mapeo de la imagen *fisheye* a la esfera unitaria utilizando la función de proyección equidistante. De este modo, en el segundo experimento (apartado 5.7.2), descrito en el actual apartado, se evalúa esta propuesta y se compara frente a un experimento donde no se utiliza este paso de corrección, es decir, considerando únicamente proyección equidistante, y empleando el modelo de cámara de Scaramuzza et al. [10]. En el segundo experimento también se analiza la calidad de las vistas derivadas de la utilización de la transformación propuesta, la cual corresponde a un polinomio, y se compara con la matriz afín.

Relacionado con los dos experimentos, la tabla 5.2 muestra las diferentes variaciones del algoritmo, en cuanto a la representación en formato esférico se refiere. Como se puede ver, a cada una de estas opciones se le ha asignado un identificador (columna Abrev.) para poder reconocer con cuál de ellas se ha generado la vista esférica completa que se está tratando. En el caso de la proporcionada directamente por la Garmin VIRB 360, este identificador será VIRB.

Tabla 5.2: Tabla con información relevante para el apartado de experimentos y resultados. Se enumeran las distintas variaciones del algoritmo en función de los pasos o métodos escogidos para la proyección esfera unitaria/imagen *fisheye* durante la transformación a formato esférico.

Opción	Abrev.	Método	Ecuaciones
(I)	MCS	Modelo de Cámara propuesto por Scaramuzza et al. [10] (apartado 5.3.1.2.1)	Ec. (5.6), Ec. (5.7) y Ec. (5.8)
(II)	PE	Proyección equidistante (apartado 5.3.1.2.2)	Ec. (5.9), Ec. (5.10) y Ec. (5.11)
(III.1)	EFP+θ	Proyección equidistante (apartado 5.3.1.2.2) + paso de corrección de la coordenada polar θ (apartado 5.5.2)	Ec. (5.9), Ec. (5.21) y Ec. (5.11)
(III.2)	EFP+θ+r	Proyección equidistante (apartado 5.3.1.2.2) + paso de corrección de las dos coordenada polares (θ y r) (apartado 5.5.2)	Ec. (5.9), Ec. (5.21) y Ec. (5.22)

5.7.1 Experimento 1: Estudio de los métodos de proyección de la imagen *fisheye* a la esfera unitaria

En este primer experimento, el objetivo es analizar y comparar la vista completa proporcionada por la cámara Garmin VIRB 360 y aquellas obtenidas como resultado de ejecutar el algoritmo configurado para las opciones (I) modelo de cámara propuesto por Scaramuzza et al. [10] (identificado como MCS) y (II) proyección equidistante,

identificado como PE) que se describen en la tabla 5.2. En otras palabras, en este primer experimento no se analiza el paso de corrección propuesto en este capítulo ni la transformación basada en polinomio, pues las vistas completas evaluadas en este primer experimento se generaron escogiendo la opción de la matriz afín en el proceso de registro. En resumen, se evalúan tres tipos de vistas esféricas completas: (1) las proporcionadas directamente por la cámara Garmin VIRB 360 (en ocasiones nos referiremos a ella como VIRB), (2) las generadas seleccionando el modelo de cámara propuesto por Scaramuzza et al. [10] (MCS) como procedimiento de mapeo esfera/imagen *fisheye* y (3) las generadas seleccionando la proyección equidistante (PE) como procedimiento de mapeo esfera/imagen *fisheye*.

Aparte de lo comentado, este experimento se divide en dos estudios. Por un lado, se evalúan los tres tipos de vistas esféricas completas comentadas en el párrafo anterior (asociadas a una misma pose) respecto al número de marcas ArUco identificadas correctamente en una de las zonas de solape (apartado 5.7.1.1). Por otro lado, en el apartado 5.7.1.2, se realiza una evaluación de estos tres tipos de vistas completas más exhaustiva, empleando el método propuesto en el apartado 5.4.3.

5.7.1.1 Estudio basado en marcas ArUco

Un alineamiento incorrecto puede producir que aparezca información visual por duplicado al fusionar o, por el contrario, que falte. Esto puede conllevar a que la identificación de un objeto en la zona de solape sea un gran desafío o incluso que sea imposible de conseguir. Este efecto es apreciable para los seres humanos pues tenemos cierto conocimiento de la forma o tamaño de las cosas, sin embargo, para no depender de la percepción humana, en este primer estudio, se propone situar la cámara de forma que en la zona de solape se capture un mosaico compuesto por distintas marcas ArUco (ver figura 5.14(a)). Así, a modo de evaluación inicial del proceso de combinación, podemos comprobar cuántas marcas ArUco son identificadas para cada una de las vistas completas a evaluar.

Para esta evaluación inicial, se colocaron 18 marcas ArUco con distintos identificadores, formando un mosaico tal y como se muestra en la figura 5.14(a). Después, se posicionó la cámara de forma que dicho mosaico fuese capturado en la zona de solape. En esa misma posición se adquirió una vista completa proporcionada por la Garmin VIRB 360 y también un par de imágenes *fisheye*.

Con el par de imágenes *fisheye*, se ejecutó dos veces el algoritmo. Por un lado, se generó una vista completa seleccionado la proyección mediante el modelo de cámara de Scaramuzza et al. [10] (MCS) y la matriz afín. Por otro lado, se generó otra vista completa pero esta vez seleccionado la proyección equidistante (PE) y la matriz afín. Estas dos vistas completas generadas junto con la proporcionada por la Garmin VIRB 360 serán las que se evaluarán en este estudio.

En este estudio nos centramos en la zona de solape en la que aparecen las marcas ArUco, así en la figura 5.14 se muestran las regiones de las imágenes a estudiar. Además, las marcas que han sido identificadas correctamente se encuentran señalas con un recuadro verde. Analizando esto, se han identificado un total de 6 marcas en la vista completa proporcionada por la Garmin VIRB 360 (figura 5.14(b)). Cabe señalar

que corresponden a las que se encuentran más alejadas de la zona de solape. En el caso del modelo de cámara de Scaramuzza et al. [10] (figura 5.14(c)), todas las marcas han sido detectadas. A diferencia del caso de la proyección equidistante (figura 5.14(d)) en el que se identificaron un total de 16 marcas ArUco, las que no fueron detectadas son aquellas que se encuentran en las zonas problemáticas (con mayor distorsión), es decir, la parte superior e inferior.

(a) (b) (c) (d)

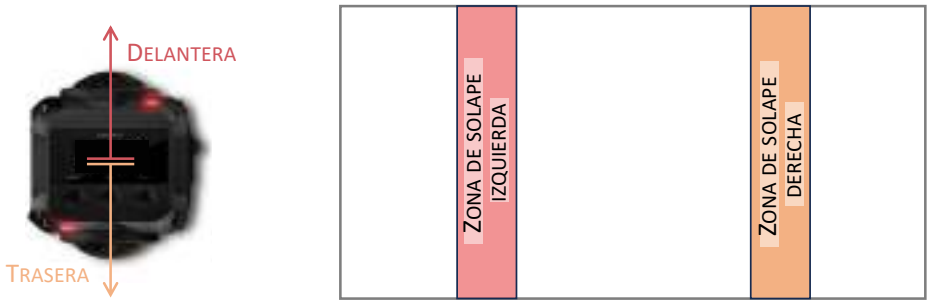
Figura 5.14: Estudio sobre el número de marcas ArUco que aparecen en (a) son identificadas, y por tanto aparecen resaltadas en color verde, en la zona de solape de la vista completa (b) proporcionada por la Garmin VIRB 360, (c) generada con el modelo de cámara de Scaramuzza et al. [10] y (d) generada con la proyección equidistante.

En resumen, la mejor combinación se produce empleando el modelo de cámara de Scaramuzza et al. [10] para el mapeo de 3D a 2D, ya que así se consigue identificar todas las marcas presentes en la zona de solape. Por contra, el peor resultado es el obtenido con la vista completa proporcionada por la Garmin VIRB 360. A pesar de esto, este estudio no es suficiente para concluir que se ha mejorado la vista completa con respecto a la que proporciona la propia Garmin VIRB 360, pero sí para tener una conclusión inicial. Con objeto de lograr una conclusión más general y que se adapte a más situaciones (como una zona de solape menos rica en detalle), se han realizado más experimentos cuyos resultados se analizan en los siguientes apartados.

5.7.1.2 Estudio basado en diferencia de descriptores de apariencia global

Dado que el método que se emplea en este experimento es el descrito en el apartado 5.4.3, se requiere de una región de referencia para comparación, la cual debe

contener la misma información visual del entorno pero sin que se haya aplicado el proceso de fusión. De este modo no tendrá efectos asociados a este paso (apartado 5.3.3). Teniendo en cuenta esto y lo comentado en el párrafo anterior, se sugirió rotar $90°$ la Garmin VIRB 360 respecto al eje perpendicular al suelo (ver figura 5.16(a)) pero manteniendo la misma posición en la que se capturaron las imágenes *fisheye* y la vista completa a evaluar (ver figura 5.15(a)). Al realizar esto, la información de la escena que antes aparecía en la zona problemática (región de solape izquierda y derecha, aproximadamente en los $-90°$ y los $90°$ de longitud respectivamente, como se pueden visualizar en la figura 5.15(b)) ya no lo hace. Ahora cada una de estas zonas ha sido capturada únicamente por una de las cámaras y, en la vista completa tomada con esa nueva orientación (figura 5.16(a)), aparece en la parte central (cuando la longitud es $0°$) y a ambos lados, como se pueden visualizar en la figura 5.16(b).



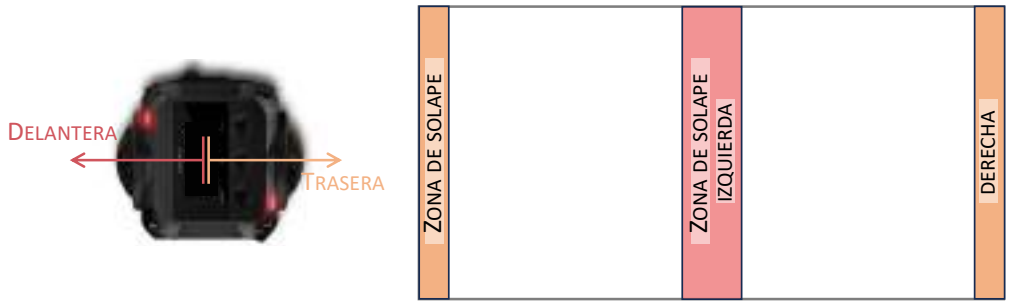
(a) (b)

Figura 5.15: Vista esférica a evaluar. Situando la cámara como se muestra en (a), se adquiere el par de imágenes *fisheye* así como la vista completa que proporciona la cámara al establecer el modo 360. La vista esférica completa que se conseguirá tendrá el aspecto mostrado en (b). En consecuencia, las zonas de solape se encuentran centradas en, aproximadamente, un $1/4$ (zona solape izquierda) y $3/4$ (zona solape derecha) de la anchura de la vista completa.

Después de obtener la región de solape a evaluar y su correspondiente región de referencia, se aplica el algoritmo de la figura 5.8 dos veces por cada par de imágenes, una para evaluar la zona de solape derecha y otra para evaluar la zona de solape izquierda.

Por consiguiente, durante la captura de la base de datos para este primer experimento, además de adquirir el par de imágenes *fisheye* y la vista completa configurando la cámara en modo 360 en cada posición, manteniendo esta última configuración, se rotó la cámara mediante el procedimiento detallado y se capturó la vista completa que se escogerá como referencia. Para este experimento no se emplearon todas las imágenes del *dataset* descrito en el apartado 5.6 (tabla 5.1), únicamente se utilizó una parte de este. Concretamente, las 24 capturadas en la estancia oficina, las 6 capturadas en la estancia laboratorio, y las 2 capturadas en el pasillo de la planta 0. Así pues, este primer experimento se ha realizado con imágenes en un total de 32 ubicaciones.

Así, para cada una de las 32 posiciones, se tiene una vista completa proporcionada por la Garmin VIRB 360 (VIRB) y otras dos como resultado de ejecutar el algoritmo (figura 5.2) descrito en el apartado 5.3 empleando la matriz afín para la etapa de registro y dos de las variaciones implementadas durante la transformación de las imágenes



(a) (b)

Figura 5.16: Vista esférica de referencia. La cámara, la cual se encuentra con la pose que se muestra en 5.15(a), se rota aproximadamente 90° alrededor de su eje vertical, obteniendo una nueva pose (a). Al capturar ahora una de las zonas de solape aparece en el centro de la vista completa mientras que la otra zona de solape se encuentra partida en los laterales, como se puede apreciar en (b).

fisheye a equirectangular: proyección sobre la esfera con (I) el modelo de cámara de Scaramuzza et al. [10] (MCS) (descrito en el apartado 5.3.1.2.1) y (II) proyección equidistante (PE) (descrito en el apartado 5.3.1.2.2).

Por consiguiente, se extraerán ambas zonas de solape (izquierda y derecha) de cada una de las tres vistas completas a evaluar (VIRB, MCS y PE) y cada una de ellas, junto con la de referencia, son la entrada al algoritmo mostrado en la figura 5.8, obteniendo una medida de diferencia que corresponde a la distancia entre los descriptores.

Tabla 5.3: Resultados generales de los los valores de distancia entre descriptores obtenidos con el algoritmo mostrado en la figura 5.8.

Tipo de vista completa	Zona de solape Izda.		Zona de solape Dcha.	
	Media [↓]	Desviación	Media [↓]	Desviación
VIRB	0.0620	0.0388	0.0534	0.0332
MCS	**0.0572**	0.0397	**0.0486**	0.0224
EP	0.0762	0.0400	0.0621	0.0306

En la tabla 5.3 se muestra la media aritmética así como la desviación estándar para cada tipo de vista completa que se quiere evaluar (VIRB, MCS y EP) y cada una de las dos zonas de solape (Izda. y Dcha.). Teniendo en cuenta que estos valores corresponden a distancias entre descriptores globales, cuanto menor sea este valor (en la tabla se indica con el símbolo [↓]) más parecido será a la imagen de referencia y por ende tendrá mayor calidad de imagen. Así, de forma general, podemos concluir que las zonas de solape correspondientes a utilizar la proyección equidistante (PE) durante la transformación a equirectangular tienen una menor calidad según el método descrito en el apartado 5.4.3. Sin embargo, hay que tener en cuenta que este método de evaluación se basa en un descriptor de apariencia global, además de que la referencia que se toma es la misma para las tres evaluaciones (VIRB, MCS y EP) y se adquirió con la cámara Garmin VIRB 360. Analizando las tres vistas completas, podemos observar que la

proyección entre la proporcionada por la cámara VIRB 360 y la generada utilizando el modelo de cámara de Scaramuzza (MCS) son similares, al contrario que la vista completa generada usando la proyección equidistante (PE). Esto se puede ver en la figura 5.17. Si nos fijamos en el armario con la boca de incendio, que corresponde a la zona de solape, se observa cómo su tamaño es muy similar tanto en la figura 5.17(b), generada por la Garmin VIRB 360, como en la figura 5.17(c), generada a partir del modelo de cámara de Scaramuzza et al. [10] (MCS). Por el contrario, su tamaño es mayor (ocupando una mayor área de la imagen) en la figura 5.17(d).

Atendiendo a todo esto, en el caso de la vista completa con PE, una distancia entre descriptores mayor no tiene porqué corresponder a una peor calidad de la zona de solape ya que en este caso la proyección difiere de la de referencia pudiendo conllevar a que de forma global no sean parecidas aún teniendo buena calidad. Por dicho motivo, en el segundo experimento (apartado 5.7.2) se volverá a comparar y evaluar estas tres vistas completas (junto con las generadas usando el paso de corrección propuesto) con otras medidas de calidad para así tener más información antes de concluir que este tipo de vista completa (EP) es de peor calidad que las otras dos.

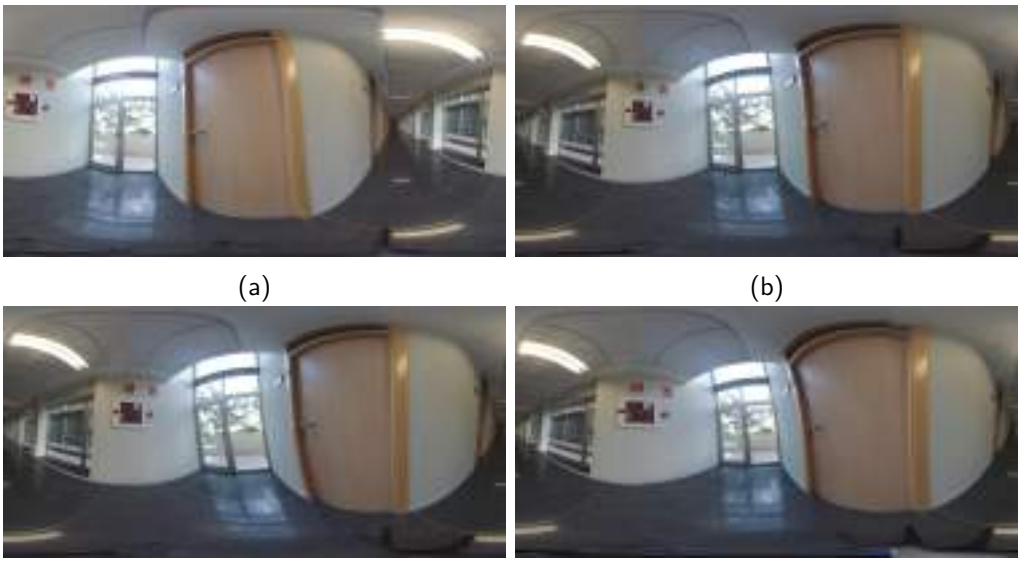
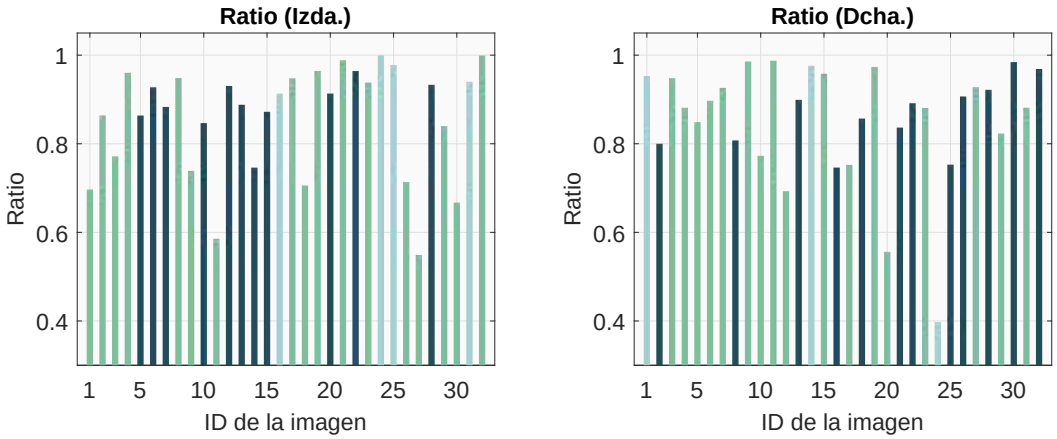


Figura 5.17: Ejemplo de vistas completas en el pasillo de la planta 0 (ver tabla 5.1). Se muestra (a) la vista completa que se toma como referencia y las tres que se van a evaluar: (b) VIRB, (c) MCS y (d) EP. En el armario de la boca de incendio se puede observar que el tamaño en (b) y (c) es bastante similar mientras que en (d) aparece ampliado.

Dado que, como se ha comentado, la apariencia entre la vista completa proporcionada por la VIRB y la generada usando el modelo de cámara de Scaramuzza et al. [10] son similares, sí que podemos concluir que las vistas completas generadas con este último procedimiento presentan una mayor calidad que las primeras (VIRB). Esto se debe a que, aun habiéndose capturado las de referencia con el mismo sistema (i.e. la Garmin VIRB 360) que las primeras, los valores medios de distancia obtenidos para ambas zonas de solape usando el modelo de cámara de Scaramuzza et al. [10] son los más bajos.

Después de esta evaluación general en la que se ha visto que se puede conseguir una vista completa de mayor calidad si en vez de capturarla directamente con la cámara VIRB 360 se captura el par de imágenes *fisheye* y se genera la vista completa con el algoritmo descrito en el apartado 5.3 seleccionando el modelo de cámara de Scaramuzza et al. [10] durante la transformación a equirectangular, también vamos a estudiar cada una de las posiciones individualmente analizando la ratio entre la menor distancia (vista con mayor calidad) y la segunda distancia menor (vista con calidad intermedia). Además, como tanto la evaluación general (tabla 5.3) como el estudio basado en marcas ArUco han establecido que la mejor calidad se consigue en las vistas completas con MCS, después se estudia, para esta configuración del algoritmo, el número de pares de correspondencias con las que se ha estimado la matriz afín así como su distribución en el eje y de la zona de solape.

Atendiendo al primer aspecto, tras obtener las tres distancias, se ordenan en orden ascendente de forma que la primera distancia corresponderá a la vista completa con mayor calidad mientras que la tercera distancia corresponderá a la vista completa con menor calidad. En la figura 5.18 se muestra la ratio entre la primera y la segunda menor distancia (eje vertical en las gráficas) para cada una de las 32 ubicaciones de la cámara (eje horizontal en las gráficas), tanto con la zona de solape izquierda (figura 5.18(a)) como la derecha (figura 5.18(b)).



(a) (b)

Figura 5.18: Ratio entre la primer y la segunda menor distancia obtenidas tras ejecutar tres veces el algoritmo de la figura 5.8 siendo una de las entradas el área de referencia y la otra entrada la zona de solape (a) izquierda y (b) derecha de cada una de las tres vistas completas a evaluar (VIRB, MCS y PE). El color determina a qué vista completa (VIRB ■, MCS ■ y PE ■) corresponde la menor distancia.

Al analizar tanto la figura 5.18(a) como la figura 5.18(b), podemos ver cómo la menor distancia (i.e. mayor similitud a la de referencia) se ha conseguido un mayor número de veces (concretamente en 17 ocasiones para ambas zonas de solape) al realizar la transformación a equirectangular con el modelo de cámara de Scaramuzza et al. [10] (MCS). La vista proporcionada por la cámara VIRB 360 ha obtenido una distancia menor a los otros dos tipos de vistas 11 veces en el caso de la zona de solape izquierda y 12 en el de la derecha. Por el contrario, tanto la zona izquierda como

la derecha de las vistas completas generadas con la proyección equidistante (PE) ha adquirido una menor distancia en muy pocas ocasiones, 4 y 3 veces, respectivamente. En cuanto a los valores de ratios calculados, estos se encuentran aproximadamente entre 0.4 y 1. Si nos fijamos en aquellos que tienen un valor relativamente bajo (cercano o menor a 0.7), lo que significa que la diferencia con respecto a la segunda distancia menor es considerable, se han conseguido con el modelo de cámara de Scaramuzza et al. [10] (MCS), excepto en el caso del valor más bajo (0.4) que corresponde a la proyección equidistante (PE).

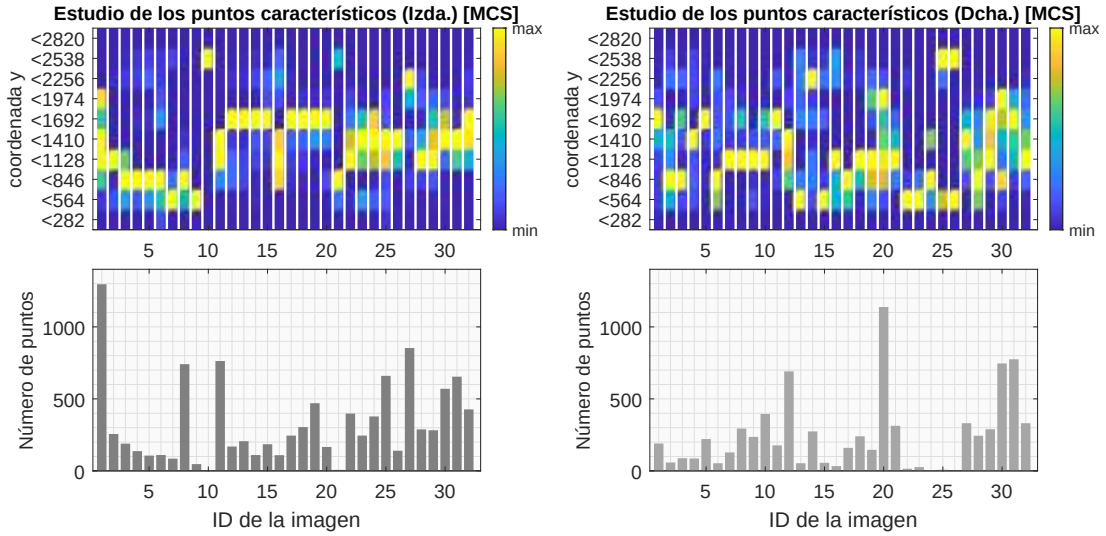
Centrándonos en los resultados cuando la transformación a equirectangular se lleva a cabo con el modelo de cámara de Scaramuzza et al. [10] (MCS), la figura 5.19 muestra el análisis de este tipo de vista completa con respecto a dos aspectos, correspondiendo la figura 5.19(a) a la zona de solape izquierda y la figura 5.19(b) a la zona de solape derecha.

Por un lado, se analiza la distribución de los pares de correspondencias en la zona de solape izquierda/derecha con respecto a su coordenada *y*, esto se muestra en la gráfica superior de la figura 5.19(a) y la figura 5.19(b), respectivamente. Para visualizar esta distribución, la zona de solape se divide en 10 regiones a lo largo de su eje y. De este modo, cada región corresponde a un intervalo de valores de la coordenada y y se contabiliza que el número de puntos característicos, que forman parte del conjunto de correspondencias, se encuentra en cada una de estas regiones. En esta gráfica, el eje vertical representa cada uno de estos 10 intervalos. Cada intervalo aparecerá con un color distinto en función del número de puntos característicos contabilizados. Dicho color viene establecido por una escala en la que el color amarillo significa que, en dicho intervalo (o parte de la altura), se encuentran ubicados el mayor número de pares de correspondencias, mientras que, al otro extremo de la escala de color se encuentra el color azul y por lo tanto significa todo lo contrario, es decir, el menor número de pares de correspondencias.

Por otro lado, también se estudia el número de pares de correspondencias, cuyos resultados se muestran en las gráficas inferiores de la figura 5.19(a) y la figura 5.19(b). Si analizamos estos resultados de forma conjunta con la distribución (gráficas superiores) y la información acerca del tipo de vista completa con menor distancia que se visualizan en la figura 5.18, podemos concluir que, como era de esperar, este tipo de vista completa (MCS) tiene una calidad más alta cuando el número de puntos característico es elevado y se encuentran distribuidos a lo largo del eje y (vertical) de la zona de solape (como ocurre en las zonas de solape izquierda correspondientes a los índices 1 y 11). Por contra, con un número de puntos característicos bajo (poca textura en la zona de solape), la vista completa derivada de usar el modelo de cámara de Scaramuzza et al. [10] tiene una calidad menor e inferior a la de la cámara Garmin VIRB 360 (como ocurre en la zona de solape derecha correspondientes al índice 25).

5.7.2 Experimento 2: Evaluación de los resultados al implementar el paso de corrección

En este apartado, se han empleado varias medidas de calidad de la imagen (IQ) para evaluar las vistas completas obtenidas tras aplicar el paso de corrección entre el par de imágenes *fisheye* que se ha propuesto en este trabajo y que se ha descrito en



(a) (b)

Figura 5.19: Estudio acerca de los pares de correspondencias encontradas durante la etapa de registro entre el par de imágenes equirectangulares tras emplear el modelo de cámara de Scaramuzza et al. [10]: (a) distribución de la coordenada y de estos puntos en la zona de solape y (b) número de pares de correspondencias.

el apartado 5.5. En lo que respecta a esta implementación, son dos las variaciones que se analizan en este apartado. Por un lado, se realizan únicamente la corrección de la coordenada polar angular (θ), que se identificará en las gráficas como PE+θ, y, por otro lado, se corrigen ambas coordenadas polares, es decir, tanto el ángulo (θ) como la distancia radial (r), así pues, esta variación se representará mediante PE+θ+r.

Como se verá a continuación, el objetivo de este segundo experimento no solo es evaluar y comparar las vistas completas generadas tras implementar estas dos variaciones, sino también compararlas con las obtenidas tras emplear los métodos analizados en el experimento anterior. En otras palabras, se evaluarán todos los casos que se describen en la tabla 5.2.

En cuanto a las medidas de calidad, en este segundo experimento, se utiliza, por un lado, una medida de nitidez de la zona de solape (IQ Nitidez) la cual se describió en el apartado 5.4.1 y cuyos resultados se muestran y analizan en el apartado 5.7.2.1. Por otro lado, se calcula la medida de calidad de imagen conocida como MS-SSIM, la cual se describió en el apartado 5.4.2 y cuyos resultados se muestran y analizan en el apartado 5.7.2.2. Destacar que esta medida únicamente se ha aplicado a las zonas de solape correspondientes a las vistas generadas con las variaciones que aparecen en la tabla 5.2 (es decir, no en las de las vistas proporcionadas por la Garmin VIRB 360) y además, se han combinado estas variaciones de tipos de proyección sobre la esfera con los dos tipos de transformación geométricas descritos en el apartado 5.3.2.

Además de las medidas de calidad de imagen que se han mencionado, en este segundo experimento, también se ha realizado una evaluación en cuanto a otros aspectos como la distancia entre pares de correspondencias de las imágenes a fusionar (es decir, el par de imágenes equirectangulares tras aplicar la transformación obtenida en la etapa

de registro), así como el tiempo de cálculo para cada combinación del algoritmo. Los resultados dcorrespondientes al primer aspecto se mostrarán y estudiarán en el apartado 5.7.2.3, mientras que el tiempo de cálculo se analiza en el apartado 5.7.2.4.

Con todo lo mencionado, se realiza una evaluación cuantitativa de las zonas de solape. Sin embargo, para poder realizar una evaluación más exhaustiva, también se ha llevado a cabo una evaluación cualitativa en el apartado 5.7.2.5, donde se muestran varias zonas de solape y se analizan de forma visual.

Cabe señalar que en este segundo experimento se han utilizado las imágenes capturadas en todas las estancias del *dataset* (ver tabla 5.1) a excepción del pasillo de la planta 0.

5.7.2.1 Estudio basado en una medida de calidad de la imagen sin referencia

En primer lugar, se ha realizado un estudio de la calidad de las zonas de solape de las vistas completas, el cual se basa en una medida que no requiere una imagen de referencia, como ocurría con la descrita en el primer experimento. Dado que no requiere una imagen de referencia, permite evaluar también la vista completa proporcionada por la cámara Garmin VIRB 360. Así pues, se compara la calidad de esta con la de las vistas completas generadas tras ejecutar el algoritmo descrito con cada una de las variaciones para representar las imágenes *fisheye* en formato esférico.

Para esta primera evaluación, se estudia la zona de solape en el dominio de la frecuencia mediante la transformada de Fourier y se obtiene una medida (ecuación (5.17)) que determina cómo de nítida es la zona de solape (IQ Nitidez). Se establece que la imagen será más nítida, y como consecuencia tendrá mayor calidad cuanto mayor sea este valor, ya que significará que tiene una mayor cantidad de componentes de alta frecuencia. Esto se indica en las gráficas de la figura 5.9 con el símbolo $\uparrow$ en el título del eje vertical que es el asociado a dicha medida.

En cuanto a este estudio, la figura 5.20 muestra los resultados de estas medidas dividiéndose en función de la zona de solape (izquierda primera columna y derecha segunda columna) pues se evalúa cada una de ellas de forma independiente, y además se han dividido en función de la estancia en la que se capturaron las imágenes. En cada una de estas gráficas, el eje horizontal representa el tipo de vista completa que se está evaluando mientras que el eje vertical corresponde a la medida de calidad de la imagen (IQ) basada en la nitidez. Para cada uno de los tipos de vistas completas se muestran todos los valores obtenidos mediante marcadores circulares y su color establece la desviación de esta medida de calidad con respecto a la media aritmética (que se representa mediante una línea horizontal azul) de todos los valores obtenidos para este tipo de vista en dicha escena. Asimismo, debajo del eje horizontal de cada gráfica aparecen unos porcentajes que representan la desviación en tanto por ciento del valor medio en cada tipo de vista completa con respecto al valor medio obtenido más alto, en otras palabras, con respecto al método con mayor calidad en función de los valores medios calculados.

Analizando los resultados de la figura 5.20, observamos que, según los valores medios, en todas las gráficas las vistas completas generadas con el modelo de cámara de Scaramuzza et al. [10] (MCS) son las que presentan un valor de nitidez más alto.

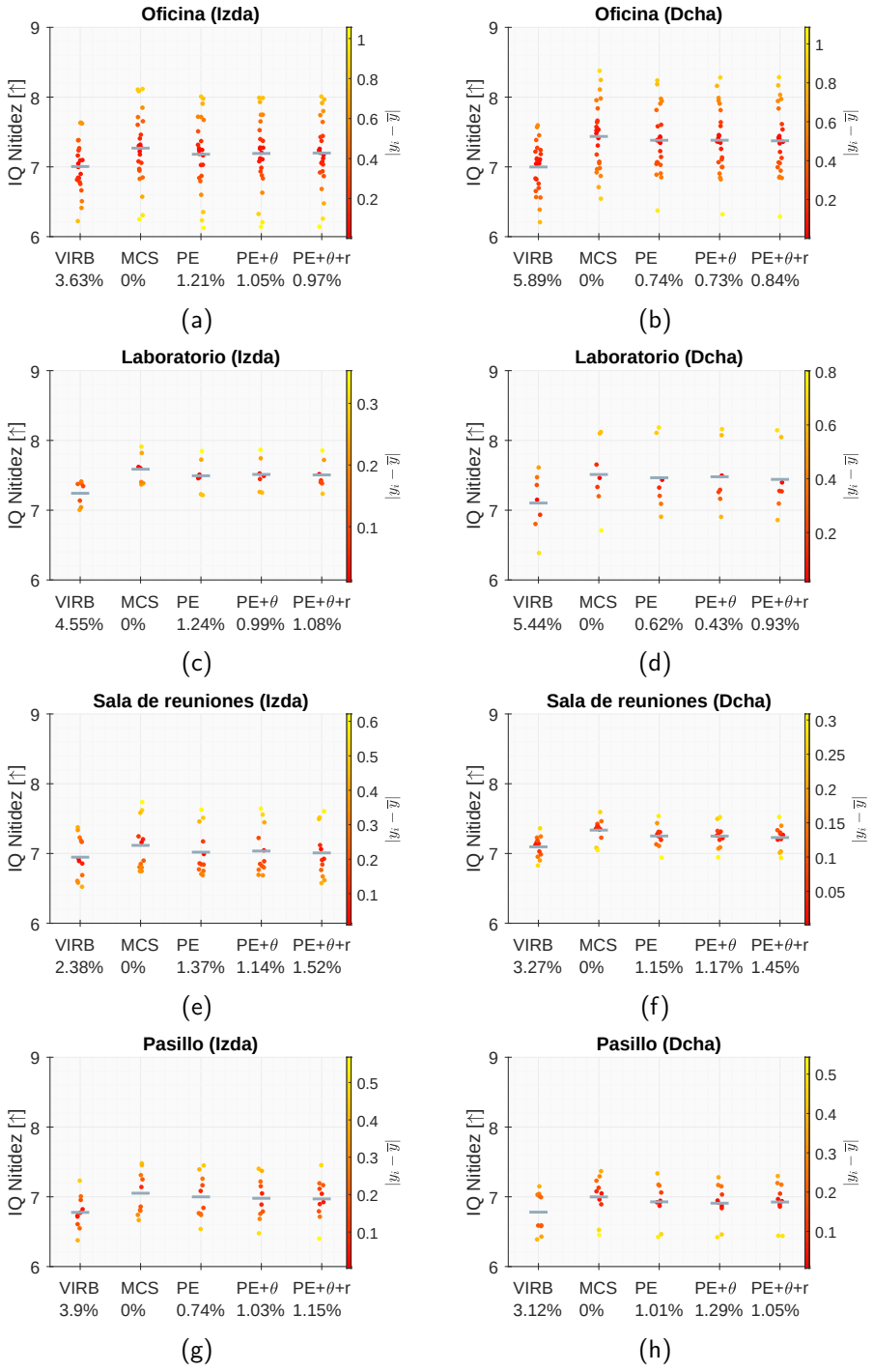


Figura 5.20: Estudio de la calidad de las zonas de solape de la vista completa en función de la nitidez (apartado 5.7.2.1). Las gráficas de la primera columna, i.e. (a), (c), (e) y (g), corresponden a la zona de solape izquierda (Izda) mientras que la segunda columna, i.e. (b), (d), (f) y (h), a la derecha (Dcha). Se comparan las diferentes variaciones descritas en la tabla 5.2 y la proporcionada por la cámara VIRB. En las gráficas, se representan todos los valores obtenidos de esta medida (y_i) mediante marcadores circulares cuyo color viene dado por su diferencia con la media aritmética ($\overline{y}$) que se muestra con una línea horizontal.

De hecho, se puede ver cómo el porcentaje siempre es cero, mientras que las vistas completas proporcionadas por la cámara (VIRB) presentan los peores resultados. En cuanto a esto último, la desviación del valor medio de este tipo de vista completa con respecto a la mejor (MCS) se encuentra entre $2.38\,\%$ (figura 5.20(e)) y un $5.89\,\%$ (figura 5.20(b)), lo que implica desviaciones considerables, sobre todo si las compramos con el resto de tipos de vistas que no alcanzan el $1.5\,\%$.

Como se ha comentado, los resultados están divididos en diferentes gráficas en función de la estancia. Esto nos ayuda a analizar si existe una relación entre esta medida de calidad y la cantidad de información visual. Así, las zonas de solape correspondientes a las estancias más ricas en textura, como son la oficina (figura 5.20(a) y figura 5.20(b)) y el laboratorio (figura 5.20(c) y figura 5.20(d)), alcanzan valores más altos de esta medida de calidad, incluso alguna zona de solape alcanza o supera el valor de 8.

5.7.2.2 Estudio basado en una medida de calidad de la imagen con referencia

En segundo lugar, se ha realizado un estudio de la calidad de las zonas de solape, pero solo de aquellas vistas completas generadas a partir del par de imágenes *fisheye* (no de la proporcionada por la Garmin VIRB 360), pues la medida de calidad escogida para este segundo estudio requiere de una imagen de referencia. En este estudio se evalúa la calidad de las zonas de solape en función de, concretamente, la medida MS-SSIM.

En el primer experimento (apartado 5.7.1.2), se empleó un método basado en imagen de referencia y se evaluó también la VIRB. Sin embargo, como se comentó, el procedimiento para obtener esa imagen de referencia desfavorece a la proyección equidistante. Dado que la principal contribución que se evalúa en este segundo experimento (el paso de corrección) sigue dicha proyección se ha optado por no utilizar la medida MS-SSIM para evaluar la proporcionada por la cámara. Así, ahora la imagen de referencia se extrae de una de las imágenes equirectangulares antes del registro, pues ahí todavía no se ha combinado la información visual y, por ende, no hay efectos debido a dicho proceso.

Además de evaluar los tipos de vistas completas según el procedimiento escogido durante la conversión del par de imágenes *fisheye* a equirectangulares, también se evalúan las dos alternativas durante el registro de imágenes: matriz afín y polinomio.

Respecto a los resultados obtenidos en este estudio, estos se presentan en la figura 5.21. Al igual que en el estudio anterior, se dividen en función de la estancia así como de la zona de solape. En cada una de las gráficas, para cada tipo de vista completa hay dos cajas de diagramas: una para los resultados con la matriz esencial (con un tono morado) y otra para los resultados con el polinomio como transformación geométrica (con un tono verde). Aparte de estas cajas, también se visualizan los valores obtenidos mediante marcadores circulares que codifican, mediante el color, su desviación respecto al valor medio, que se representa mediante una línea horizontal gruesa sobre las cajas. Cabe recordar, tal y como se indica en el título del eje vertical de las gráficas mediante $\uparrow$, que cuanto mayor sea el valor obtenido, mayor será la calidad.

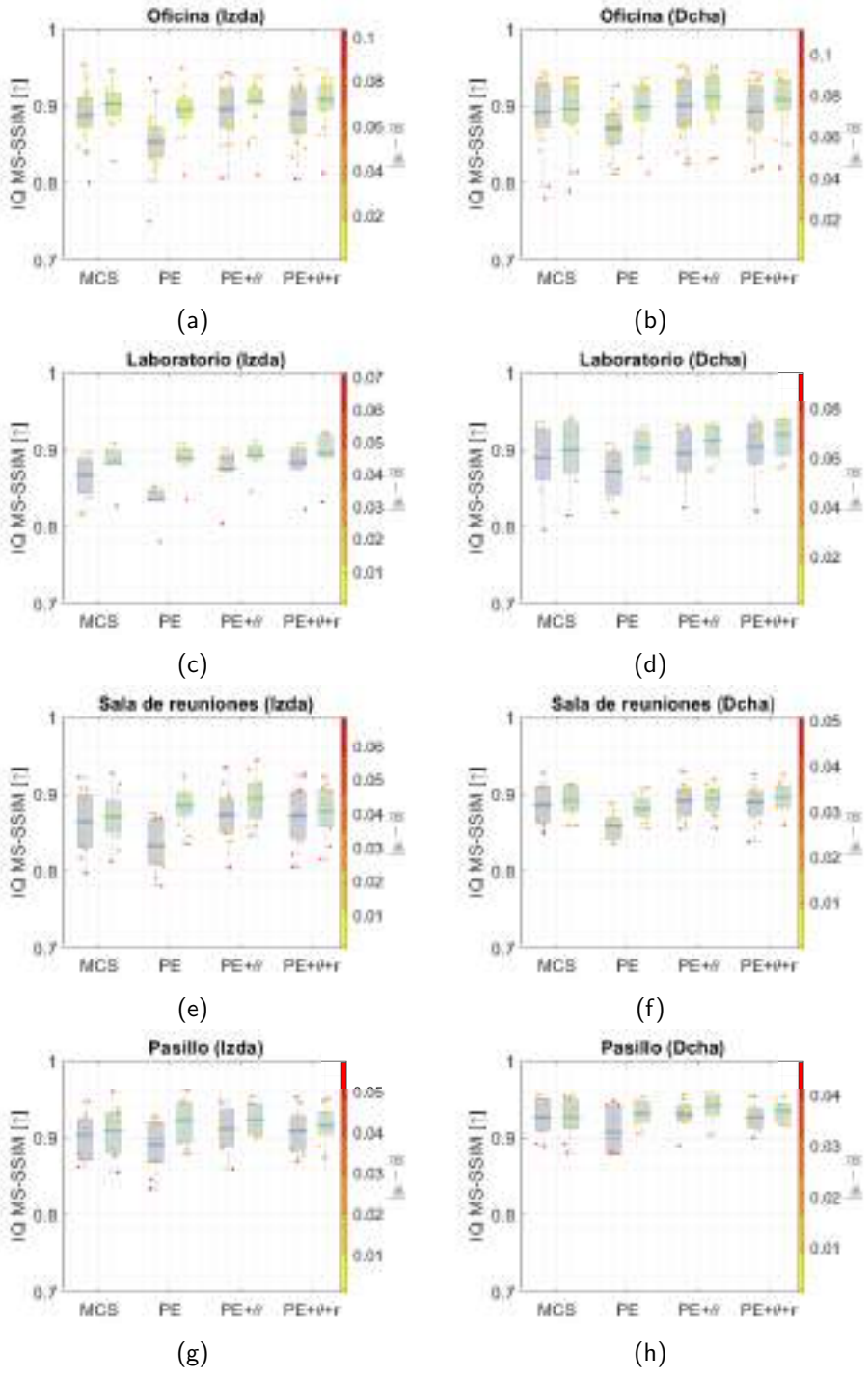


Figura 5.21: Estudio de la calidad de las zonas de solape de la vista completa calculando la medida MS-SSIM (apartado 5.7.2.2). Como se indica mediante ↑ en el título del eje vertical de las gráficas, cuanto mayor sea este valor mayor será la calidad de la imagen. Las gráficas de la primera columna, i.e. (a), (c), (e) y (g), corresponden a la zona de solape izquierda (Izda) mientras que la segunda columna, i.e. (b), (d), (f) y (h), a la derecha (Dcha). Se comparan las diferentes variaciones descritas en la tabla 5.2 utilizando una matriz afín ( ) y un polinomio ( ).

5.7.2.3 Estudio de la distancia entre pares de correspondencias en el plano imagen equirectangular

Los dos estudios anteriores se basan en efectos perceptibles como la nitidez (apartado 5.7.2.1) o la luminancia, el contraste y la estructura (apartado 5.7.2.2). Si nos centramos en aquellos artefactos producidos por un incorrecto registro del par de imágenes equirectangulares, podemos decir que, en esta etapa, el objetivo es que los píxeles, que son la proyección de un mismo punto 3D de la escena, tengan la misma posición antes de ejecutar la etapa de fusión de estas, tal y como se representa visualmente en la figura 5.22.

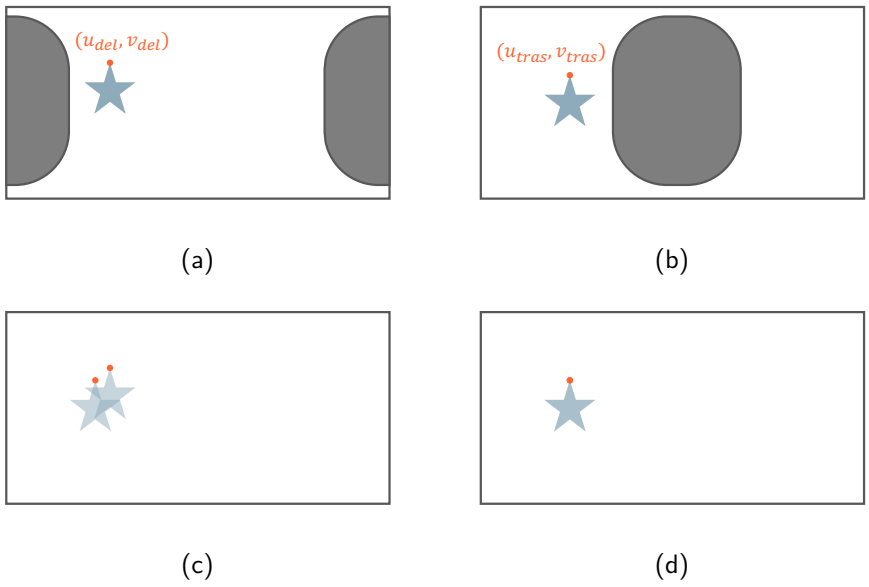


Figura 5.22: Dado el par de imágenes equirectangulares, donde (a) es la delantera y (b) la trasera, en el caso de que las proyecciones de un mismo punto 3D no tengan las mismas coordenadas, es decir, $u_{del} \neq u_{tras}$ y $v_{del} \neq v_{tras}$, cuando se fusionan las dos zonas de solape para obtener la vista completa (c) se observa que ese punto 3D aparece por duplicado. Por el contrario, si se cumpliese que $u_{del} = u_{tras}$ y $v_{del} = v_{tras}$ (i.e. un correcto registro), tras fusionar ambas imágenes se obtendría una vista completa (d) en la que no se aprecia el efecto anterior.

De modo que, teniendo en cuenta esto, el siguiente estudio consiste en calcular la distancia en píxel entre puntos característicos detectados en las dos imágenes equirectangulares y que corresponden al mismo punto 3D (i.e. pares de correspondencias). Así pues, se espera que si el registro ha sido correcto esta distancia (o error) sea prácticamente cero. Cabe mencionar que, al igual que el estudio anterior, este únicamente se ha realizado con las variaciones descritas en la tabla 5.2 y no con el caso de la cámara Garmin VIRB 360, pues no se dispone del par de imágenes equirectangulares antes de la fusión.

En la figura 5.23, se representan los resultados de este estudio. En primer lugar, la figura 5.23(a) muestra la diferencia en píxeles entre los pares de correspondencias encontrados entre las dos imágenes equirectangulares antes de aplicar la transformación estimada durante la etapa de registro. Como se puede observar, el error medio más alto

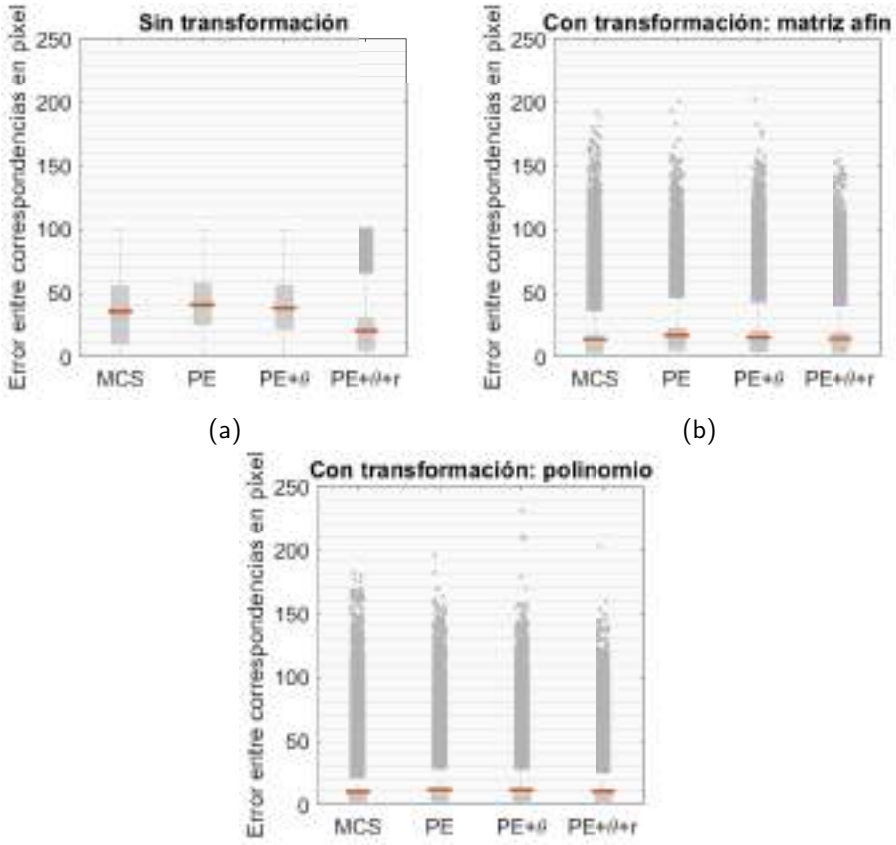


Figura 5.23: Estudio de la distancia (error) entre los pares de correspondencias encontrados para el par de imágenes equirectangulares obtenidos a la salida del (a) primer módulo del algoritmo (sin transformación), y del segundo módulo aplicando (b) una matriz afín y (c) un polinomio como transformación geométrica. La línea horizontal (_) muestra el valor medio para cada caso.

se observa cuando las imágenes equirectangulares se obtienen mediante la proyección equidistante. Por el contrario, el menor valor medio corresponde al uso de las dos correcciones propuestas en este trabajo (PE+θ+r). Respecto a las otras dos variaciones restantes, con el modelo de cámara de Scaramuzza et al. [10] (MCS) el error medio es menor que realizando únicamente la corrección de la coordenada polar θ.

En resumen, atendiendo a lo comentado en la figura 5.22 y lo mostrado en la figura 5.23, se puede decir que los resultados de este estudio determinan que si tras representar en formato equirectangular el par de imágenes *fisheye* se fusionasen, sin realizar la etapa de registro, la vista completa que menos artefactos presentaría debido a un registro poco preciso se daría si se realiza el paso de corrección propuesto en apartado 5.5.2, tanto para la coordenada angular (θ) como para la radial (r).

En segundo lugar, la figura 5.23(b) está asociada al par de imágenes equirectangulares tras estimar la matriz afín en la etapa de registro y aplicarla a una de ellas. En esta gráfica se puede ver cómo la distancia entre los pares de correspondencias ha disminuido para todas las variaciones del algoritmo con respecto a no aplicar ninguna transformación geométrica. En cuanto a cada uno de los tipos, el comportamiento es

muy similar al anterior, ya que el valor de distancia media más alto corresponde a utilizar la proyección equidistante sin la corrección propuesta en este trabajo, seguido de esta misma proyección, pero corrigiendo la coordenada polar θ. Por último, tanto aplicar la corrección a ambas coordenadas polares (r y θ) como utilizar el modelo de cámara de Scaramuzza et al. [10] (MCS) presentan los mejores resultandos, aunque se podría decir que el comportamiento del primero de ellos es el deseable pues los *outliers* se encuentran más cerca del valor medio.

En tercer lugar, la figura 5.23(c) corresponde al par de imágenes equirectangulares tras estimar el polinomio en la etapa de registro y aplicar dicha transformación geométrica a una de ellas. Si comparamos los resultados generales de esta gráfica con las dos anteriores, observamos que con este tipo de transformación no lineal se logra una distancia media menor que con una transformación lineal (matriz afín) y, como era de esperar, un valor aún más pequeño respecto a no aplicar ninguna. En lo que respecta a cada uno de los tipos de procedimiento para convertir a formato equirectangular, el comportamiento es el mismo que utilizando la matriz afín, siendo los mejores resultados los del modelo de cámara de Scaramuzza et al. [10] (MCS) y la proyección equidistante con la corrección de ambas coordenadas polares y el peor caso cuando se emplea solo la proyección equidistante.

5.7.2.4 Estudio del tiempo computacional

Pese a que la calidad de la vista completa es importante, también lo es el tiempo que tarda en ejecutarse el algoritmo para conseguirla, especialmente si el objetivo es utilizarla en tiempo real, como sería en el campo de la robótica. Como consecuencia, en este apartado se estudia este aspecto.

Tras ejecutar el algoritmo de la figura 5.2 para cada uno de los pares de imágenes *fisheye* combinando el procedimiento para transformar estas imágenes a formato equirectangular (ver tabla 5.2: MCS, PE, PE+θ o PE+θ+r) y los dos tipos de transformaciones geométricas en la etapa de registro (matriz afín o polinomio), se ha calculado el tiempo medio para cada una de estas combinaciones cuyos valores se definen en la tabla 5.4.

Tabla 5.4: Tiempo computacional medio tras ejecutar el cada versión del algoritmo con todos los pares de imágenes *fisheye* del *dataset*.

Transformación geométrica	Tiempo medio en segundos [↓]. MCS	PE	PE+θ	PE+θ+r
Matriz afín (apartado 5.3.2.1)	368.88 s	**36.02** s	36.87 s	37.59 s
Polinomio (apartado 5.3.2.2)	375.48 s	**36.77** s	37.70 s	38.35 s

Como puede verse, el tiempo computacional es menor cuando se establece la matriz afín como transformación geométrica en vez de utilizar un polinomio para el registro del par de imágenes, pese a que la diferencia es menor a un segundo en las variaciones de proyección equidistante (PE, PE+θ y PE+θ + r) y menor a siete segundos en el caso del modelo de cámara de Scaramuzza et al. [10] (MCS).

En lo que respecta a la configuración de la etapa de transformación a formato equirectangular, el empleo de la proyección equidistante (PE) hace que se genere la vista completa final más rápidamente que en el resto de los casos. Mientras que si se corrige al menos una de las coordenadas polares (θ) se emplea un tiempo mayor (sin llegar a un segundo) y, como era de esperar, este tiempo aumenta un poco más si se corrigen las coordenadas polares (θ y r). Aunque utilizar la proyección equidistante sin implementar el paso de corrección propuesto sea la opción más rápida, no es la más adecuada pues según los estudios realizados en este segundo experimento, esta opción presenta la peor calidad, mientras que esta se mejora al realizar la corrección propuesta y la diferencia en cuanto a tiempo es muy pequeña.

Si bien hemos visto que utilizar el modelo de cámara de Scaramuzza et al. [10] (MCS) proporciona una mayor calidad en la vista completa final, tal y como muestran los resultados de la tabla 5.4, esto se cumple a costa de un mayor tiempo computacional (prácticamente diez veces más que el tiempo del resto de casos). Igualmente, hay que añadir el tiempo que conlleva realizar previamente el paso de calibración de ambas cámaras.

5.7.2.5 Valoración cualitativa

Para finalizar este segundo experimento, se propone complementar los estudios cuantitativos anteriores con una evaluación cualitativa, pues teniendo en cuenta que cada una de las medidas se centran en este aspecto, puede que no corresponda con lo que los humanos percibimos de forma visual. Así pues, a continuación, se muestran algunos ejemplos de las zonas de solape evaluadas para analizar su calidad en función del criterio humano.

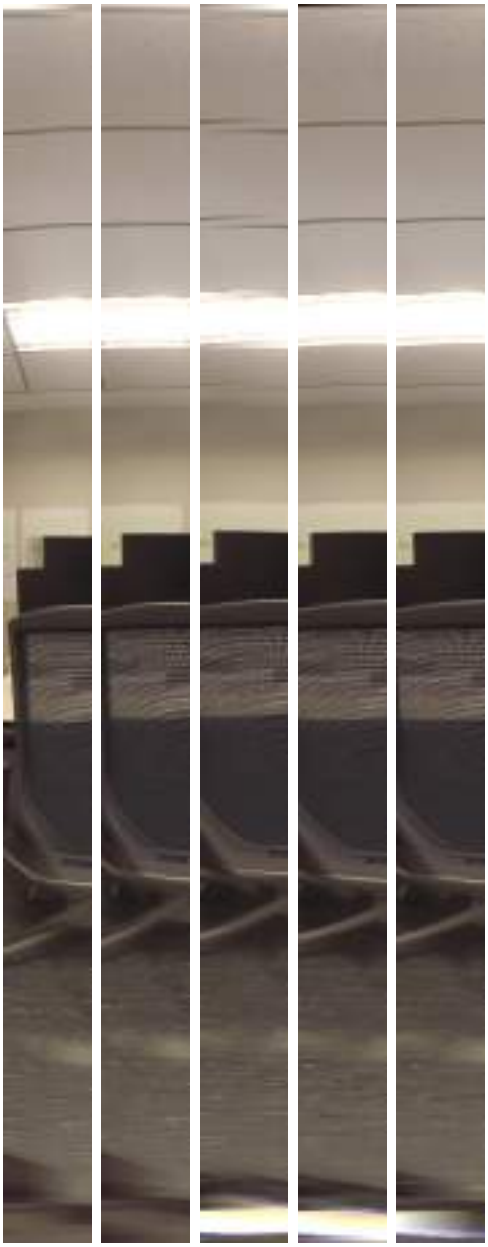
En primer lugar, las figuras 5.24, 5.25, 5.26 y 5.27 muestran las zonas de solape de las cinco vistas completas que se han evaluado en el primer estudio de este segundo experimento: (a) VIRB, (b) MCS, (c) PE, (d) PE+θ y (e) PE+θ+r.

Analizando cada una de estas figuras de forma independiente, en la figura 5.24 podemos ver las zonas de solape en las que aparece el mosaico de marcas ArUco utilizado en el apartado 5.7.1.1. En este ejemplo podemos apreciar cómo el paso de corrección mejora la calidad de la vista completa, pues si nos fijamos en la marca superior e inferior del mosaico, que eran las que no se podían identificar en el caso de proyección equidistante, tras corregir ambas coordenadas polares (figura 5.24(e)) los resultados mejoran apreciablemente. Esto también se puede ver en el ejemplo de la figura 5.25, en el caso de la proyección equidistante sin corrección (figura 5.25(c)), observando la parte del techo y el póster de la pared así como las líneas horizontales que no están alineadas ni en vertical. Si nos fijamos en la pantalla del ordenador, el borde vertical izquierdo tampoco está alineado, existe un pequeño desplazamiento horizontal. Si se corrige la coordenada polar θ (figura 5.25(d)) se observa que se consigue que las líneas horizontales (techo y póster) estén alineadas, sin embargo esto no ocurre con las líneas verticales (borde izquierdo de la pantalla). Esto último se logra corrigiendo también la otra coordenada polar (figura 5.25(e)).

El ejemplo mostrado en la figura 5.26 corresponde a una zona de solape pobre en textura. Como se concluyó en uno de los estudios, en estos casos (pocos puntos

(a) (b) (c) (d) (e)

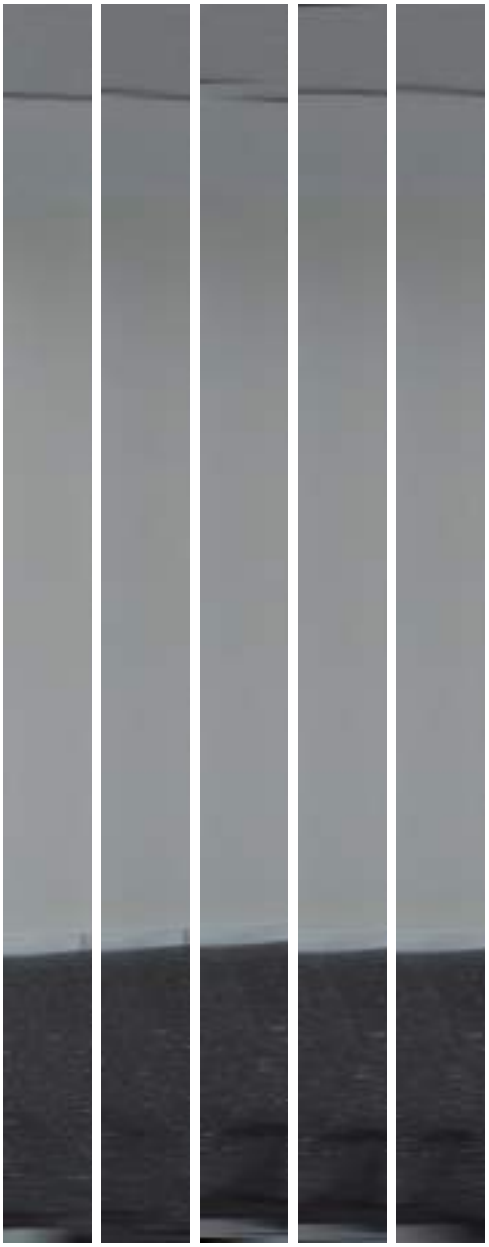
Figura 5.24: Captura en oficina. Zonas de solape a evaluar (posición 1, zona de solape izquierda): (a) VIRB, (b) MCS, (c) PE, (d) PE+θ y (e) PE+θ+r.



(a) (b) (c) (d) (e)

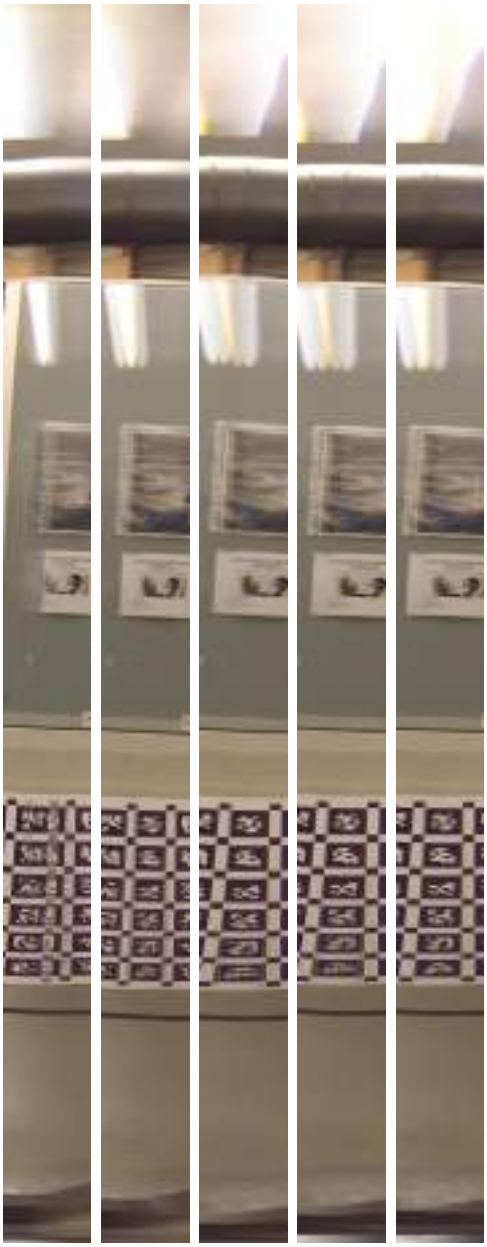
Figura 5.25: Captura en oficina. Zonas de solape a evaluar (posición 6, zona de solape derecha): (a) VIRB, (b) MCS, (c) PE, (d) PE+θ y (e) PE+θ+r.

característicos), la zona de solape que visualmente se aprecia mejor corresponde a la proporcionada por la cámara VIRB 360 (figura 5.26(a)), ya que la línea en la zona superior es más nítida. Pese a haber poca cantidad de información visual relevante, aquí también se puede ver cómo, tras realizar cualquiera de las dos correcciones, PE+θ (figura 5.26(d)) o PE+θ+r (figura 5.26(e)), se mejora la proyección equidistante (figura 5.26(c)). Esto se puede ver en la zona superior correspondiente al techo. En este



(a) (b) (c) (d) (e)

Figura 5.26: Captura en pasillo. Zonas de solape a evaluar: (a) VIRB, (b) MCS, (c) PE, (d) PE+θ y (e) PE+θ+r.



(a) (b) (c) (d) (e)

Figura 5.27: Captura en el laboratorio. Zonas de solape a evaluar: (a) VIRB, (b) MCS, (c) PE, (d) PE+θ y (e) PE+θ+r.

caso, relacionándolo con el análisis del ejemplo anterior, corregir una sola coordenada polar o las dos se comporta prácticamente igual, pues solo hay líneas horizontales que ya se corregían con θ.

En cuanto al último ejemplo mencionado, figura 5.27, aquí también se puede observar, sobre todo en el patrón de calibración AprilTag, cómo se mejora la calidad al implementar el paso de corrección. Entre las dos variaciones del paso de corrección, la mejora es mayor en la segunda (PE+α+r), ya que, como se observa en la figura

5.27(e), en este caso se puede ver la última marca con mayor nitidez, cosa que no ocurre con las otras zonas de solape.

Para finalizar esta evaluación cualitativa, la figura 5.28 muestra las zonas de solape utilizando la matriz afín durante el proceso de registro, mientras que la figura 5.29 corresponde al alineamiento del par de imágenes equirectangulares (proceso de registro) mediante un polinomio de grado dos. Tanto la figura 5.28 como la figura 5.29 están referidas a la misma ubicación. Si comparamos ambas figuras, podemos ver cómo utilizar el polinomio para este ejemplo mejora la calidad, pues se observa una mayor nitidez en el cartel que aparece en la zona superior.

5.8 Software implementado para la generación de vistas completas

Además de haber realizado los diferentes estudios y evaluaciones, también se ha diseñado un software que dado un par de imágenes *fisheye* genera una vista esférica completa siguiendo el algoritmo descrito en este trabajo, con todas las variaciones mencionadas (incluidas las aportaciones). En la figura 5.30, se muestra la interfaz de este software programado con Matlab.

Para su uso, principalmente, se debe seguir el orden establecido por el algoritmo descrito a lo largo de los apartados anteriores. Así pues, en primer lugar, el usuario tiene que cargar el par de imágenes *fisheye* clicando en el botón "*Load dual fisheye images*". Antes de comenzar con el primer módulo del algoritmo, se deben configurar varios parámetros y seleccionar qué opción se pretende ejecutar. En cuanto a los parámetros, como puede verse en la parte central izquierda de la interfaz (figura 5.30), se pide que se introduzca el campo de visión de ambas lentes en grados (en nuestro caso $201.8°$) y que se defina el tamaño (altura y anchura) de la vista esférica completa, guardando la relación de 2:1. Además, también es posible definir la rotación relativa entre las cámaras introduciendo los grados para la rotación alrededor de cada eje. Por defecto esta rotación es la ideal, es decir, la que se muestra en la figura 5.4(a).

Respecto a las opciones, por defecto la proyección para mapear desde la esfera a la imagen *fisheye* es la equidistante (apartado 5.3.1.2.2), en cuyo caso podemos seleccionar si queremos que se realice el paso de corrección propuesto en este trabajo (apartado 5.3.1.2.1). En caso afirmativo, también se debe seleccionar si se quiere corregir únicamente una de las coordenadas polares o las dos (tal y como se ha realizado en el segundo experimento). En el caso de que el método que se quiera aplicar para este paso sea el modelo de cámara de Scaramuzza et al. [10], el apartado de corrección aparecerá deshabilitado (pues únicamente se ejecuta con proyección equidistante) y, aparte de esto, para poder ejecutar este método se tienen que cargar los ficheros con los parámetros de calibración correspondientes a la cámara delantera (botón "*Front camera*") y a la trasera (botón "*Back camera*"). Tras definir todo esto, se procede a ejecutar el primer módulo del algoritmo clicando en el botón "*Convert to spherical format*". El par de imágenes equirectangulares que son la salida de este primer módulo se muestran en la parte superior derecha de la interfaz (al lado del par de imágenes *fisheye*). Pulsando con el botón derecho sobre una de estas imágenes, el usuario puede guardarlas o abrirlas en una nueva figura.

Tal y como indica el algoritmo de la figura 5.2, para generar la vista esférica

(a) (b) (c) (d) (e)

Figura 5.28: Captura en sala de reuniones. Zonas de solape a evaluar tras usar la matriz afín: (a) VIRB, (b) MCS, (c) PE, (d) PE+θ y (e) PE+θ+r.

(a) (b) (c) (d) (e)

Figura 5.29: Captura en sala de reuniones. Zonas de solape a evaluar tras usar el polinomio: (a) VIRB, (b) MCS, (c) PE, (d) PE+θ y (e) PE+θ+r.

completa se ejecuta el proceso de registro y fusión. En la interfaz, ambos módulos se ejecutan simultáneamente por lo que se deben configurar los parámetros de ambos de forma conjunta. Respecto al proceso de registro, se puede seleccionar el tipo de puntos característicos con los que se va a calcular la transformación geométrica. Aunque en este trabajo se ha empleado ORB, en la interfaz también se ha dado la opción de SURF. Además de esto, se puede seleccionar el tipo de transformación geométrica:

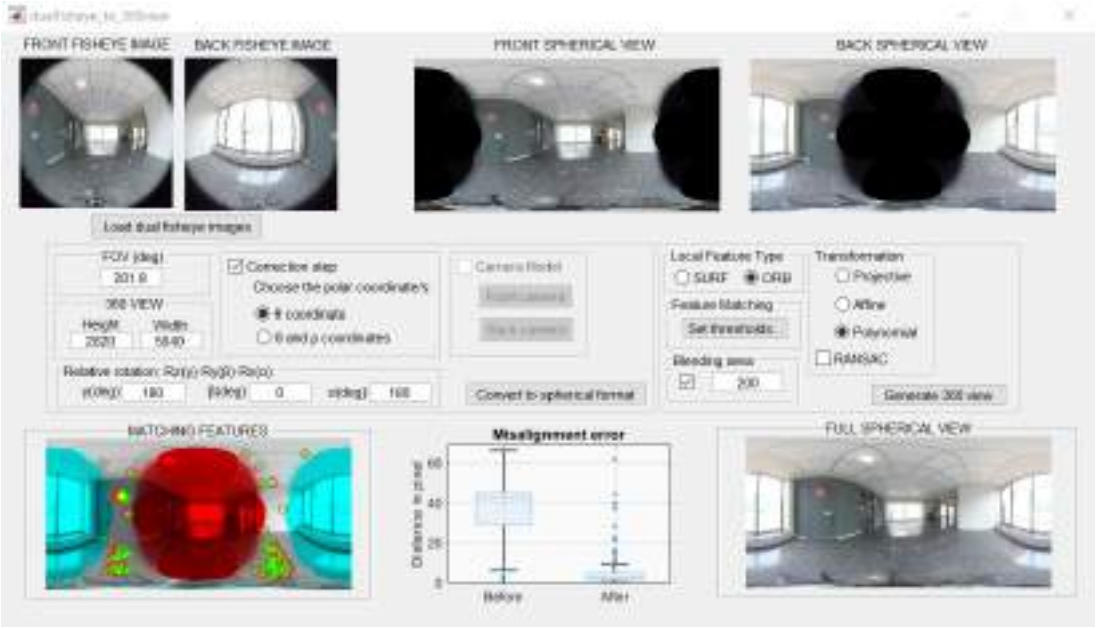


Figura 5.30: Interfaz del *software* diseñado para generar vistas completas a partir de un par de imágenes *fisheye*.

proyectiva, afín o polinómica. En cuanto al proceso de fusión, se ofrece configurar el ancho de la zona de solape o si se quiere que este se estime de forma automática. Una vez configurado todo esto, se clica en "*generate 360 view*" para que se genere la vista esférica completa, la cual se abre en una nueva figura, pero también se muestra en la parte inferior derecha. Además, también se muestran, de forma visual, los pares de correspondencias con los que se ha obtenido la transformación entre el par de imágenes esféricas (proceso de registro) y una gráfica en la que se puede ver la distribución de las distancias obtenidas entre los pares de correspondencias entre el par de imágenes equirectangulares antes del proceso de registro y después.

5.9 Conclusión

El presente capítulo ha tenido como objetivo generar una vista completa a partir de un par de imágenes *fisheye* que tenga una calidad mayor a la que proporciona automáticamente la misma cámara, Garmin VIRB 360.

El procedimiento para generar una vista completa a partir de un par de imágenes *fisheye* se compone principalmente de tres módulos. En primer lugar, el par de imágenes de entrada son transformadas a un formato esférico, generalmente el equidistante. En segundo lugar, se lleva a cabo el registro del par de imágenes, ahora en formato esférico, para estimar cuál es la relación entre ellas pues el objetivo es que, en el plano equirectangular, los píxeles que son la proyección de puntos 3D de la escena que son comunes a ambas (i.e. se encuentran en el campo de visión compartido) deben tener la misma ubicación en ambas imágenes antes de la fusión de estas pues lo contrario puede ocasionar ciertos efectos visuales indeseables. Esto principalmente ocurre porque cada par de imágenes de entrada al algoritmo ha sido capturada por un sensor distinto

y además el centro no es común, dado que existe un cierto desplazamiento así como una rotación entre ambos sistemas de referencia. En tercer lugar, las dos imágenes equirectangulares son fusionadas en una única que corresponderá a la vista completa.

En este capítulo, hemos mostrado este algoritmo, pero con ciertas alternativas en los dos primeros módulos con la finalidad de poder evaluar la calidad de las vistas completas resultantes y así escoger aquella combinación que, según los resultados, genere la de mejor calidad. Por un lado, dado que, para convertir una imagen *fisheye* a un formato esférico 2D (primer módulo), el procedimiento consta de un mapeo desde 2D a 3D, siendo esta última sobre la superficie de una esfera unitaria, seguida de un mapeo desde 3D a 2D, donde el plano corresponde a la imagen esférica, se requieren dos tipos de proyecciones. Para este último mapeo de esfera unitaria (3D) a plano imagen esférica (2D) hemos seleccionado la proyección equirectangular que es de las más empleadas. Sin embargo, para el primer mapeo, hemos implementado dos opciones: (I) el modelo de cámara de Scaramuzza et al. [10] y (II) la proyección equidistante. Tras realizar un estudio entre los pares de correspondencias en el plano imagen *fisheye*, utilizando esta segunda opción, percibimos cierta relación que podría modelarse utilizando una función seno y un factor de escala. Esto se ha implementado en el algoritmo durante la transformación a formato esférico como un paso de corrección previo. Así, la segunda opción tiene tres variantes: (II) solo proyección equidistante, (II.1) proyección equidistante aplicando el paso de corrección propuesto para una de las coordenadas (θ) y (II.2) proyección equidistante aplicando el paso de corrección propuesto para corregir ambas coordenadas (θ y r). Además, durante el estudio realizado, también se observó cierta no linealidad entre las coordenadas cartesianas de las correspondencias entre los pares de imágenes equirectangulares antes del módulo de registro. Este hecho nos hizo plantear la estimación e inclusión de una transformación geométrica no lineal durante el segundo módulo del algoritmo (i.e. registro de imágenes), concretamente proponemos emplear un polinomio de grado 2.

Además de lo mencionado con respecto a la generación de una vista completa, también hemos propuesto un método automático para evaluar la calidad de la zona de solape, el cual se basa en comparar dos descriptores holísticos adquiridos como la salida de la primera capa *fully-connected* (FC6) de la red neuronal VGG16. Uno de ellos corresponde a la zona de solape (izquierda o derecha) de la vista completa que se quiere evaluar y el otro descriptor a esa misma área de la imagen, pero sin efectos debido a la unión (i.e. zona de solape de referencia). Para adquirir esta última, como se ha mencionado en el capítulo, se ha rotado la cámara 90° con respecto a su eje vertical para que la información capturada en la zona de solape en la pose anterior aparezca ahora capturada sólo por una de las dos lentes.

En cuanto a los experimentos, primero se realizó uno en el que se evalúan y comparan los siguientes tres tipos de vistas completas: las generadas por la propia cámara, las obtenidas utilizando el modelo de cámara de Scaramuzza et al. [10] y utilizando la proyección equidistante. Todo ello empleando el método basado en redes neuronales comentado en el párrafo anterior. De los resultados obtenidos en este experimento, podemos determinar que se consigue una vista completa con mayor calidad si empleamos el algoritmo con el modelo de cámara de Scaramuzza et al. [10] que si configuramos la cámara de forma que proporcione esta vista directamente en vez del par de imágenes

fisheye. Aparte de esto, se observó que el método basado en redes neuronales propuesto es una alternativa de evaluación de la calidad de la zona de solape bastante aceptable, aunque dicha medida se acerca más a la realidad cuando el tipo de proyección entre la imagen de referencia y la que se desea evaluar es bastante similar; esto se analiza en la figura 5.17. En cambio, esto no ocurre entre la vista completa establecida como referencia y la vista completa generada con la proyección equidistante.

Después, se realizó el segundo experimento en el que también se comparan los tres tipos de vistas completas del primer experimento más las dos variaciones correspondientes al paso de corrección propuesto. Además, también se evaluaron y compararon las dos transformaciones geométricas comentadas: matriz afín y polinomio. En este experimento, se utiliza una medida de calidad de la imagen sin referencia basada en la nitidez y otra medida que sí requiere de una imagen de referencia. A su vez, se evalúan todos los tipos de vista que se pueden generar con el algoritmo (excluyendo la proporcionada por la cámara) analizando la distancia entre las correspondencias encontradas en los pares de correspondencias. La conclusión de este segundo experimento es que el paso de corrección que proponemos mejora la calidad de la vista completa resultante con respecto a no implementarlo (únicamente proyección equidistante) incluso llegando a igualar la del empleo del modelo de cámara de Scaramuzza et al. [10] sin tener que realizar un proceso previo de calibración.

5.9.1 Trabajos futuros

En el primer experimento se realizó una primera evaluación en función de cuántas marcas ArUco se podían detectar en la zona de solape. Siguiendo esta línea, pero con un nivel superior, se propone como trabajo futuro utilizar todos los tipos de vistas completas (incluidas las de la cámara) para resolver tareas como reconocimiento y detección de objetos o personas en dichas zonas de solape, con una base de datos mucho más amplia.

Por otro lado, también se propone buscar una alternativa de imagen de referencia que sea más general para que el método basado en redes neuronales para la evaluación no beneficie a aquella que es más similar en cuanto a proyección y distinga más aspectos de la calidad de la imagen.

Por último, se intentará mejorar el algoritmo para mantener la calidad aun cuando la zona de solape es pobre en información visual y se seguirá trabajando en reducir el tiempo de cálculo de los algoritmos.

5.10 Publicaciones en las que se basa este capítulo

Los principales resultados que se han mostrado y discutido en este capítulo se han publicado en:

- M. Flores, D. Valiente, J. J. Cabrera, O. Reinoso y L. Payá, "Generation and Quality Evaluation of a 360-degree View from Dual Fisheye Images", en Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics - ICINCO, INSTICC, SciTePress, 2022, págs. 434-442, isbn: 978-989-758-585-2. doi: 10.5220/0011275900003271.

  - En este artículo se propone un método automático para evaluar la calidad de una vista completa obtenida a partir de un sistema de visión cuya configuración son dos lentes *fisheye* ubicadas *back-to-back*. Este método de evaluación consiste en dos redes neuronales que comparten pesos, siendo la entrada a una de ellas la zona de solape que se quiere evaluar y la entrada a la otra es la zona de solape de referencia. Para cada una de estas entradas se tiene un descriptor que corresponde a la salida de la primera capa *fully-connected*. Finalmente, la medida de calidad es la distancia entre el par de descriptores.

- M. Flores, D. Valiente, A. Gil, O. Reinoso y L. Payá, "Creación y análisis de vistas 360 a partir de un par de imágenes fisheye", en XLIII Jornadas de Automática: libro de actas: 7, 8 y 9 de septiembre de 2022, Logroño (La Rioja) 2022.a ed., Servizo de Publicacións da UDC, Comité Español de Automática y Universidad de La Rioja, sep. de 2022, págs. 985-992, isbn: 9788497498418. doi: 10.17979/spudc.9788497498418.0985.
  - En este artículo se han generado dos tipos de vistas completas a partir de un par de imágenes capturado por la cámara Garmin VIRB 360. Estos difieren en el procedimiento empleado durante la transformación a formato esférico. Además, se lleva a cabo un análisis de estas dos vistas creadas y la proporcionada de forma directa por la misma cámara, para finalmente, realizar una comparativa entre estas.
- M. Flores, D. Valiente, A. Peidró, O. Reinoso y L. Payá, "Generating a full spherical view by modelling the relation between two fisheye images", en Visual Computer. Artículo enviado, se encuentra actualmente en revisión.
  - Las contribuciones de este artículo son, por un lado, proponer un paso de corrección que modela la relación en coordenadas polares entre el par de imágenes *fisheye*. Dicha corrección se aplica durante la transformación de la imagen *fisheye* a formato esférico con la finalidad de que el desalineamiento entre el par de imágenes esféricas se minimice. Por otro lado, se propone utilizar una transformación geométrica polinómica durante el proceso de registro. Por último, se realiza una evaluación con distintas medidas y comparativa frente a diferentes vistas esféricas.

6 Conclusiones y trabajos futuros

En los capítulos anteriores se han presentado y descrito, de forma detallada, las diferentes contribuciones y resultados de este trabajo de investigación. De modo que, se resumen en este capítulo el cual se divide en dos apartados: aportaciones y conclusiones (apartado 6.1) y trabajos futuros (apartado 6.2).

6.1 Contribuciones y conclusiones

La presente tesis se centra principalmente en dos líneas de investigación. Por un lado, se ha abordado la tarea de localización de un robot móvil con una cámara omnidireccional a bordo. La técnica empleada para ello es la odometría visual. Por otro lado, se ha centrado en la generación de una vista de 360 grados a partir de un par de imágenes *fisheye* proporcionadas por una cámara *fisheye* dual *back-to-back*, concretamente la cámara Garmin VIRB 360. Para analizar los resultados de esta investigación, utilizamos la media del error de localización y el tiempo de cálculo para evaluar la eficacia de nuestras aportaciones en la línea de investigación de localización. Por el contrario, en la línea de investigación de generación de vistas de 360 grados empleamos medidas de calidad de imagen y tiempo de cálculo. El software utilizado en este trabajo de investigación es Matlab. Así, las principales contribuciones y resultados de cada capítulo se enumeran a continuación.

Capítulo 4

- Este capítulo presenta varias contribuciones al enfoque Adaptive Probability-Oriented Feature Matching (APOFM) [1]. APOFM fue propuesto por Valiente et al. [1] para lograr una búsqueda de coincidencia de características locales más robusta utilizando un modelo de rejilla 3D que codifica la probabilidad de

existencia de características locales en la escena.

- Para mejorar el algoritmo estándar APOFM, se propone una búsqueda ponderada y dinámica de correspondencia de características locales. Esta búsqueda se realiza con un umbral dinámico y otro estático, donde el umbral dinámico viene dado por el modelo de probabilidad 3D y una función, que puede ser un escalón, lineal o un cuadrática.
- Otra mejora propuesta es la implementación de una clasificación mediante un algoritmo de *K-nearest neighbour* con dos métricas diferentes para seleccionar el conjunto de puntos de características locales candidatos.
- Además de las dos contribuciones anteriores, también se propone como mejora un registro automático de falsos positivos, que puede utilizarse para evaluar la robustez del enfoque.
- Una vez implementadas estas contribuciones en el método APOFM, este se emplea en un algoritmo para estimar la pose relativa de un robot móvil entre dos instantes de tiempo consecutivos. De esta forma, se evalúan las contribuciones mencionadas analizando, entre otros aspectos, el error de localización. Asimismo, la evaluación se completa comparando las posibles combinaciones de las contribuciones mencionadas para APOFM y otros algoritmos de referencia.
- La evaluación se realiza, por un lado, con pares de imágenes captadas por una cámara catadióptrica y, por otro, con pares de imágenes adquiridas por una cámara *fisheye*. Esta evaluación comparativa proporciona información sobre el rendimiento relativo de ambos sistemas de campo de visión amplio y, por lo tanto, es útil a la hora de elegir el sistema de visión más adecuado.
- Como se ha mencionado anteriormente, la técnica APOFM se basa en una distribución de probabilidad 3D de la existencia de características locales en una escena. Por lo tanto, este capítulo también presenta una evaluación de este enfoque utilizando cuatro tipos de características locales: SURF [14], ORB [11], FAST [5] y KAZE [41].
- La primera conclusión tras los experimentos es que APOFM presenta mejores resultados que los métodos de referencia en cuanto a eficiencia y precisión, independientemente del tipo de cámara.
- Además, estos experimentos indican que la aplicación de las contribuciones mejora el método estándar APOFM. Por ejemplo, el uso de una función cuadrada para la búsqueda ponderada y dinámica de la coincidencia de características locales aumenta la precisión en la estimación de la pose relativa.
- Comparando los resultados para ambos tipos de imágenes, otra conclusión es que se obtiene un mayor número de características candidatas cuando la entrada al algoritmo es un par de imágenes *fisheye*, aunque el número de características que finalmente encuentran una coincidencia en la otra imagen (es decir, número de pares de coincidencia de características) es mayor cuando la entrada es un par de imágenes catadióptricas.
- El error de localización es menor usando imágenes catadióptricas que usando imágenes de *fisheye*, pero el comportamiento de los falsos positivos es mejor

usando imágenes de *fisheye*.

- Tras estudiar la influencia de diferentes tipos de características locales en el algoritmo APOFM con imágenes de *fisheye* como entrada, el uso de ORB proporciona mejores resultados de localización con una buena precisión durante la búsqueda de características coincidentes.

Capítulo 5

- El algoritmo para generar una vista esférica completa a partir de un par de imágenes *fisheye* consta principalmente de al menos tres etapas: transformación de las imágenes *fisheye* a formato esférico, registro y mezcla. El presente trabajo se centra en las dos primeras etapas.
- La primera contribución de este capítulo es un estudio sobre la relación entre pares de características coincidentes de (a) un par de imágenes en formato esférico y (b) un par de imágenes *fisheye*.
- A partir de los resultados del punto anterior, se propone un paso de corrección. Este paso consiste en una primera estimación de los parámetros del modelo, que relaciona las características coincidentes en coordenadas polares de un par de imágenes *fisheye*. Posteriormente, este modelo y los parámetros estimados se emplean para transformar la imagen de *fisheye* posterior al formato esférico. El objetivo es minimizar la diferencia de píxeles entre las proyecciones de los mismos puntos 3D en las imágenes esféricas.
- En cuanto a la segunda etapa del algoritmo, se propone un polinomio como transformación geométrica para registrar el par de imágenes esféricas.
- La primera etapa del algoritmo (es decir, la transformación a un modelo esférico) se compone de dos mapeos, uno de la imagen esférica a la esfera y un segundo de la esfera al plano de la imagen. Para esta última proyección, se implementan dos procedimientos: utilizando la proyección de *fisheye* equidistante o utilizando el modelo de cámara propuesto por Scaramuzza et al. [10]. Hay tres variaciones para el primero: sin el paso de corrección, con la corrección de una coordenada polar o la corrección de ambas coordenadas polares.
- Los experimentos se basan en evaluar la calidad de las diferentes imágenes esféricas completas generadas mediante el algoritmo con las opciones implementadas. Además, se comparan con la imagen esférica completa proporcionada por Garmin VIRB 360.
- Para la evaluación de la calidad de la zona de solapamiento se propone un método automático. Este método utiliza dos arquitecturas de redes neuronales convolucionales para generar dos descriptores holísticos: uno para la zona de solapamiento a evaluar y otro para la zona de solapamiento de referencia. La medida de calidad es la distancia entre ambos vectores de descripción.
- Este método automático basado en *deep learning* se utiliza para evaluar la calidad de las vistas esféricas completas generadas con los dos procedimientos implementados, que relacionan la esfera unitaria y la imagen fisheye, y la imagen proporcionada por una cámara Garmin VIRB 360. Se obtiene una puntuación de calidad elevada cuando se emplea el modelo de cámara.

- En un segundo experimento se evalúa el paso de corrección utilizando dos medidas adicionales de calidad de imagen. Los resultados muestran que la calidad de la vista esférica completa es mayor utilizando el paso de corrección que empleando únicamente la proyección de *fisheye* equidistante. Además, la calidad obtenida aplicando el paso de corrección es muy similar a la puntuación de calidad obtenida cuando se utiliza el modelo de cámara.

6.2 Trabajos futuros

Tras enumerar las aportaciones presentadas y las conclusiones extraídas en el presente trabajo, en este apartado se proponen futuros trabajos de investigación relacionados con las líneas de investigación de esta tesis.

Capítulo 4

- El algoritmo se ha diseñado para un entorno acotado, como puede ser un entorno interior. Teniendo en cuenta que el robot móvil también puede navegar en un entorno exterior, se propone como trabajo futuro adaptar el algoritmo para este último tipo de entorno.
- El segundo trabajo futuro propuesto se centra en el tipo de imagen. Una vez que el proceso para generar un 360° vista adquirida por un *back-to-back* doble cámara de *fisheye*, se propone el uso de este tipo de imágenes en el marco de localización.
- El tercer trabajo futuro propuesto se concentra en la parte del algoritmo dedicada a detectar y describir los puntos de características locales. En el presente trabajo se han evaluado diferentes métodos. Aún así, se sugiere utilizar otras características locales que sean robustas o invariantes a las propiedades de las imágenes captadas por una cámara omnidireccional.

Capítulo 5

- Un proceso de *stitching* incorrecto produce algunos efectos indeseables en la vista esférica completa. En consecuencia, los objetos de la zona superpuesta cambian de aspecto y no son fácilmente identificables. Considerando esto y que es posible que un robot móvil deba resolver tareas como reconocimiento y detección de objetos o personas durante su navegación, proponemos analizar la eficiencia de estas tareas cuando los objetos o personas aparecen en la zona de solape.
- Además, sugerimos encontrar una imagen de referencia más genérica para el método de evaluación basado en el aprendizaje profundo.
- Debido a que el algoritmo se basa en características locales, proponemos mejorarlo de forma que pueda ser utilizado en entornos no ricos en texturas.

6 Conclusions and Future Work

In the previous chapters, the different contributions and results of this research work have been presented and described in detail. Thus, they are summarised in this chapter which is divided into two sections: contributions and conclusions (section 6.1) and future work (section 6.2).

6.1 Contributions and conclusions

The present thesis has two main research lines. On the one hand, it has addressed the localization task of a mobile robot with an omnidirectional camera onboard. The technique employed for this purpose is visual odometry. On the other hand, it has focused on the generation of a 360-degree view from a pair of fisheye images provided by a back-to-back dual-fisheye camera, concretely the camera Garmin VIRB 360. To analyze the results of this research, we used the average of the localization error and the computation time to evaluate the effectiveness of our contributions in the localization research line. In contrast, we employed image quality measures and the computation time in the 360-degree view generation research line. The software used in this research work is Matlab. Thus, the main contributions and results are recapitulated as follows.

Chapter 4

- This chapter presents several contributions to the Adaptive Probability-Oriented Feature Matching (APOFM) [1] approach. APOFM was proposed by Valiente *et al.* [1] to achieve a more robust local feature matching search using a 3D grid model that encodes the probability of existence of local features in the scene.
- To improve the standard APOFM algorithm, a weighted and dynamic search of

local feature matching is proposed. This search is carried out with a dynamic and a static threshold where the dynamic threshold is given by the 3D probability model and a function, which can be a step, a line or a square.

- Another improvement proposed is the implementation of a K-nearest neighbour classification with two different metrics to select the set of candidate local feature points.
- In addition to the above two contributions, an automatic false positive record is also proposed as an improvement, which can be used to evaluate the robustness of the approach.
- Once these contributions are implemented in the APOFM method, the framework is employed in an algorithm to estimate the relative pose of a mobile robot between two consecutive time instants. In this way, the contributions mentioned above are assessed by analysing the localization error, among other aspects. Also, the assessment is completed by comparing the possible combinations of the contributions mentioned for APOFM and other baseline algorithms.
- The assessment is performed, on the one hand, with pairs of images captured by a catadioptric camera and, on the other hand, with pairs of images acquired by a fisheye camera. This comparative evaluation gives information about the relative performance of both wide field-of-view systems and therefore it is useful wheen choosing the most suitable vision system
- As mentioned above, the APOFM technique is based on a 3D probability distribution of existence of local features in a scene. Therefore, this chapter also presents an evaluation of this approach using four types of local features: SURF, ORB, FAST and KAZE.
- The first conclusion after the experiments is that APOFM presents better results than the baseline methods regarding efficiency and precision independently of the type of camera.
- In addition, these experiments indicate that the implementation of the contributions improves the APOFM standard method. For instance, the use of a square function for the weighted and dynamic search of local feature matching increases the precision estimating the relative pose.
- Comparing the results for both types of images, another conclusion is that a higher number of candidate features is obtained when the input to the algorithm is a pair of fisheye images, even though the number of features that finally find a match in the other image (i.e. number of feature matching pairs) is higher when the input is a pair of catadioptric images.
- The localization error is lower using catadioptric images than using fisheye images, but the behaviour of false positives is better using fisheye images.
- After studying the influence of different types of local features in the APOFM algorithm with fisheye images at input, the use of ORB provides better results for localization with a good precision during the search of matching features.

Chapter 5

- The algorithm to generate a full spherical view from a pair of fisheye images consists mainly of at least three stages: transformation of the fisheye images to spherical format, registration, and blending. The present work focuses on the first two stages.
- The first contribution of this chapter is a study about the relationship between pairs of matched features of (a) a pair of images in spherical format and (b) a pair of fisheye images.
- From the results of the previous item, a correction step is proposed. This step consists of a first estimation of the parameters of the model, which relates the matched features in polar coordinates of a pair of fisheye images. Afterwards, this model and the parameters estimated are employed to transform the back fisheye image into the spherical format. This aims to minimise the difference in pixels between the projections of the same 3D points in the spherical images.
- Regarding the second algorithm stage, a polynomial is proposed as a geometric transformation for registering the pair of spherical images.
- The first stage of the algorithm (i.e. transformation to a spherical model) is composed of two mappings, one from the spherical image to the sphere and a second one from the sphere to the image plane. For the latter projection, two procedures are implemented: using the equidistant fisheye projection or using the camera model proposed by Scaramuzza *et al.* [10]. There are three variations for the first one: without the correction step, with the correction of one polar coordinate or the correction of both polar coordinates.
- The experiments are based on assessing the quality of the different full spherical images generated by means of the algorithm with the options implemented. Also, they are compared to the full spherical image provided by Garmin VIRB 360.
- For the quality evaluation of the overlapping zone, an automatic method is proposed. This method uses two architectures of convolutional neural networks to generate two holistic descriptors: one for the overlapping zone to be evaluated and one for the overlapping reference zone. The quality measure is the distance between both description vectors.
- This automatic method based on deep learning is used to evaluate the quality of the full spherical views generated with the two procedures implemented, which relate the unit sphere and the fisheye image, and the image provided by a Garmin VIRB 360 camera. A high quality score is achieved when the camera model is employed.
- The correction step is evaluated using two additional image quality measures in a second experiment. The results show that the quality of the full spherical view is higher using the correction step than using only the equidistant fisheye projection. Furthermore, the quality obtained by implementing the step correction is very similar to the quality score achieved when the camera model is used.

6.2 Future Work

After enumerating the contributions presented and conclusions extracted in the present work, this section proposes future research works related to the research lines of this thesis.

Chapter 4

- The algorithm has been designed for a bounded environment such as an indoor environment. Taking into account that the mobile robot can also navigate in an outdoor environment, it is proposed as future work to adapt the algorithm for the latter type of environment.
- The second proposed future work focuses on the type of image. Once the process to generate a 360° view acquired by a back-to-back dual fisheye camera has been presented, we propose using such images in the localization framework.
- The third proposed future work concentrates on the algorithm part aimed at detecting and describing local feature points. In the present work, different methods have been evaluated. Still, it is suggested to use other local features which are robust or invariant to the properties of the images captured by an omnidirectional camera.

Chapter 5

- An incorrect stitching process produces some undesirable effects in the full spherical view. Consequently, that objects in the overlapping area change their appearance and are not easily identifiable. Considering this and the fact that a mobile robot may have to solve tasks such as recognition and detection of objects or persons during its navigation, we propose to analyze the efficiency of these tasks when objects or persons appear in the overlapping area.
- Also, we suggest finding a more generic reference image for the evaluation method based on deep learning.
- Due to the fact that the algorithm is based on local features, we propose to improve it in such a way that it can be used in environments which are not rich in texture.

Engineering Applications of Artificial Intelligence 107 (2022) 104539

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Efficient probability-oriented feature matching using wide field-of-view imaging

María Flores [a,*], David Valiente [b], Arturo Gil [a], Oscar Reinoso [a], Luis Payá [a]

[a] Department of Systems Engineering and Automation, Miguel Hernandez University, Avenida de la Universidad, s/n, Elche, 03202, Spain
[b] Communications Engineering Department, Miguel Hernandez University, Avenida de la Universidad, s/n, Elche, 03202, Spain

ARTICLE INFO

Keywords:
Feature matching
Dynamic visual model
Adaptive probability-oriented feature matching
Fisheye lenses
Omnidirectional images
Visual localization

ABSTRACT

Feature matching is a key technique for a wide variety of computer vision and image processing applications such as visual localization. It permits finding correspondences of significant points within the environment that eventually determine the localization of a mobile agent. In this context, this work evaluates an Adaptive Probability-Oriented Feature Matching (APOFM) method that dynamically models the visual knowledge of the environment in terms of the probability of existence of features. Several improvements are proposed to achieve a more robust matching in a visual odometry framework: a study on the classification of the matching candidates, enhanced by a nearest neighbour search policy; a dynamic weighted matching that exploits the probability of feature existence in order to tune the matching thresholds; and an automatic false positive detector. Additionally, a comparison of performance is carried out, considering a publicly available dataset composed of two kinds of wide field-of-view images: catadioptric and fisheye. Overall, the results validate the appropriateness of these contributions, which outperform other well-recognized implementations within this framework, such as the standard visual odometry, a visual odometry method based on RANSAC, as well as the basic APOFM. The analysis shows that fisheye images provide more visual information of the scene, with more feature candidates. Contrarily, omnidirectional images produce fewer feature candidates, but with higher ratios of feature acceptance. Finally, it is concluded that improved precision is obtained when the location problem is solved by this method.

1. Introduction

In recent years, the creation of visual models of environments has received a great attention by the scientific community, due to the numerous applications it has in a variety of areas such as in mobile robotics (Harapanahalli et al., 2019; Patruno et al., 2020; Taheri and Xia, 2021; Kostavelis et al., 2016). When a robot has to operate in an 'a priori' unknown scenario (Alatise and Hancke, 2020), modelling efficiently this environment is a crucial requisite. Nowadays, vision systems sustained by computer vision and image processing techniques are widely acknowledged to this purpose. In particular, feature matching (Jiang et al., 2013; Liu et al., 2021) permits finding, modelling and tracking relevant visual information from the environment. Once the previous task is achieved, the mobile robot will be able to solve the mapping and localization problems with robustness (Hou et al., 2020).

The present work continues the research line started in Valiente et al. (2018), where the Adaptive Probability-Oriented Feature Matching (APOFM) technique is proposed to obtain a robust local feature correspondence search in presence of outliers. This method comprises a feedback loop that accounts for the existence of previous matches in the 3D space. Such information corresponds to a 3D probability distribution provided by a Gaussian Process (GP). Finally, once the local feature points are detected in the next iteration, the 3D probability distribution of features existence aids in the selection of candidate points for the definitive matching.

The APOFM can be used in many applications where feature matching is needed (e.g. object tracking Xiao et al., 2012, detection Jakubović and Velagić, 2018, mapping Zivkovic et al., 2005 and localization Wu et al., 2011 of mobile robots). Among them, we have focused on the localization problem. Sometimes the presence of dynamic elements can cause errors in the pose estimation and a robust matching framework is required. Thence, considering the benefits of the previous method in that context, its implementation in a visual odometry algorithm can improve the solution to this problem.

This work presents several improvements to the APOFM which provide a more precise localization estimation comparing to the basic APOFM (Valiente et al., 2018). The main contributions of this work are fourfold:

* Corresponding author.
E-mail addresses: m.flores@umh.es (M. Flores), dvaliente@umh.es (D. Valiente), arturo.gil@umh.es (A. Gil), o.reinoso@umh.es (O. Reinoso), lpaya@umh.es (L. Payá).

https://doi.org/10.1016/j.engappai.2021.104539
Received 10 March 2021; Received in revised form 29 October 2021; Accepted 31 October 2021
Available online 15 November 2021

(a) The matching candidates selection has been improved by means of a k-nearest neighbour classifier based on different distance metrics.
(b) The spatial probability distribution is used to perform a weighted and dynamic search of feature correspondences, under static and adaptive thresholds.
(c) An automatic false positive detector is implemented, based on the distance between pixel points and their 3D projection.
(d) An extended comparison of the efficiency of the proposal is performed, using not only a catadioptric vision system but also a fisheye one. To that end, two open-source and publicly available image datasets (Robotics and Perception Group, University of Zurich, Switzerland, 2013) have been used to benchmark the proposal with other well-acknowledged implementations such as a Standard Method (SM) (Hartley and Zisserman, 2003), SM using RANdom SAmple Consensus (RANSAC) (Nister, 2003; Scaramuzza, 2011), as well as the basic APOFM (Valiente et al., 2018).

The remainder of this paper is structured as follows. Section 2 presents an overview of related works. In Section 3, the two types of vision system and the camera model are described. The method to recover the relative pose from a pair of images is outlined in Section 4, whereas Section 5 presents how this method concretizes based on the vehicle model. In Section 6, all the steps of the improved APOFM are explained. Finally, the results achieved during the experiments are shown in Section 7. Section 8 presents the conclusions of this work.

2. Related work

Modelling the environment consists in creating a representation. Three main approaches can be found in the related literature: topological (Cebollada et al., 2019; Román et al., 2020), metric (Andert and Goormann, 2007; Liu et al., 2020) and hybrid (Yuan et al., 2018). One of the most usual representation is the occupancy grip map (Gil et al., 2015), which discretizes the environment into cells to define free or occupied (presence of an obstacle) regions. However, the classical occupancy grid approaches have some limitations such as the fact that the structural correlations between points on the map are not considered. For this reason, new techniques, such as Gaussian Process (GP) (Rasmussen and Williams, 2006), have been applied to overcome them. This learning method is a Bayesian nonparametric approach designed to solve regression and probabilistic classification problems. GP is a powerful tool to accurately identify a complex mathematical model from experimental data. Among the variety of suitable properties of GP, its main advantage is that it deals with the noise in the system, as well as with the uncertainty in the model. In O'Callaghan and Ramos (2012), the authors present an algorithm that creates a continuous occupancy representation of the environment by GP, denominated Gaussian Process Occupancy Mapping (GPOM). Ghaffari et al. (2017) extend this algorithm to create a semantic map. To this purpose, they formulate the semantic mapping as a multi-class classification problem instead of a binary classification. The GP technique is not only applied to build a model of the environment. It has recently become popular in the research community since it can be used to solve a wide range of problems in the field of robotics (Song et al., 2018; Polymenakos et al., 2020; Sun et al., 2018; Park et al., 2018; Dalla Libera et al., 2019; Nutalapati et al., 2019; Li et al., 2020). For example, Nguyen et al. (2019) employ the GP to infer remaining wall thickness at unseen pipe sections for a mobile robot which moves inside a pipeline with the objective of inspecting at the location of a break.

Both the mapping and the localization tasks can be carried out as long as the mobile robot acquires information from its environment. To this purpose, many types of sensors (e.g. sonar, lidar, encoders, global position system) can be mounted on the mobile robot. Among them, vision sensors have become a source of countless research contributions in recent years due to the several attractive features that they present, such as the richness of information captured, low weight, power consumption, size, and cost (Reinoso and Payá, 2020). Cameras are versatile since they can be utilized for navigation both in outdoor and indoor environments. Nevertheless, the most interesting advantage is the amount of information from the environment that an image contains, such as colour, luminance, shape and texture. The use of these sensors increases the scope of applications of mobile robots. The type of information provided by them not only permits solving the localization and mapping problems, but it can also be used for other tasks, for instance, road detection (Zhang et al., 2018), traffic sign recognition (Jung et al., 2016), and obstacle identification (Emani et al., 2019). The amount of information available in an image is related to the field of view of the camera that captured it. The wider the field of view is, the higher the amount of information from the environment. According to this, this type of vision systems can be classified, in broad lines, into conventional monocular or omnidirectional cameras.

Comparing to conventional monocular, omnidirectional vision systems have more advantages thanks to their wide field of view. A single image captured by this type of camera can provide a 360° view from the environment around the mobile robot (Amorós et al., 2020). Therefore, omnidirectional cameras permit obtaining exhaustive models of the environment with a reduced number of views (Payá et al., 2017). There are different alternatives to get an omnidirectional vision system (Scaramuzza, 2014; Li, 2006). The most extended ones are dioptric and catadioptric systems. These are the configurations used in the present work. The first one consists in combining a conventional camera with a shaped wide-angle lens (such as fisheye). This vision system provides a hemispherical view, so a pair of cameras pointing to opposite sides is required to acquire a full spherical view (Gao and Shen, 2017). The second way to create an omnidirectional system is the combination of a spherical (Barone et al., 2018), conic (Marcato Junior et al., 2016), hyperbolic (Boutteau et al., 2010), parabolic, or elliptic mirror and a pinhole camera.

For some applications (e.g. autonomous aerial robots), the fisheye cameras are better than the catadioptric ones since they achieve an omnidirectional coverage with lower weight (Gao et al., 2020). Nowadays, the automotive industry is very interested in providing vision perception to the drivers, concretely a 360° view around the vehicle. To that end, the vehicles are equipped with four fisheye cameras which are placed in a way that the coverage is optimized. In this application, the coverage obtained using a catadioptric vision system is less effective since the majority of information captured in the image will be sky and body car. For instance, Lee et al. (2013) have mounted four fisheye cameras (looking front, rear, left and right) on a vehicle to implement their structureless pose-graph loop-closure algorithm.

The formulation of the localization problem typically depends on the type of sensor used. It can be classified into global (i.e. global position system) or local (i.e. wheel, inertial, laser, radar, or vision systems mounted onboard) localization. In Mohamed et al. (2019), the authors provide a general overview of the state-of-the-art about the localization methods using these latter sensors.

In the case of onboard vision systems, the technique to solve the local localization problem is also known as visual odometry, which incrementally estimates the motion of an agent. The difference is that the vision-based odometry obtains the relative pose through the changes that the movement induces in the images (Fraundorfer and Scaramuzza, 2012). This way, this method overcomes the main limitations of the wheel odometry (such as wheel slippage and uneven terrain). Besides, comparing with other traditional approaches (i.e. GPS, inertial, laser and radar), visual odometry is an inexpensive and relatively accurate alternative technique that can be employed both in outdoor and indoor environments, and its use is not only limited to ground vehicles.

Depending on the process chosen to extract the information from the images, the different methods of visual odometry can be classified into feature-based, appearance-based, or hybrid approaches (Poddar et al., 2018). In Valiente García et al. (2012), a comparison between both appearance- and feature-based visual odometry methods is carried out, using omnidirectional images.

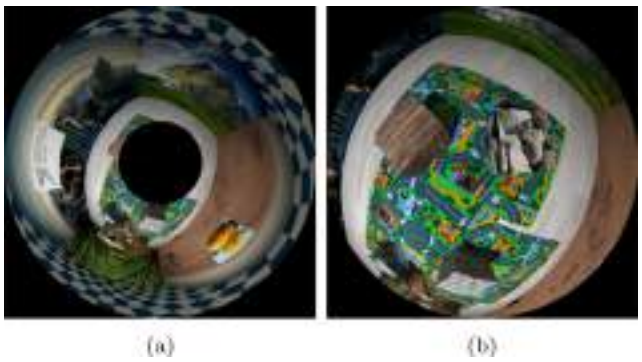


Fig. 1. Two examples of wide field of view images which are extracted in the same indoor scenario (Robotics and Perception Group, University of Zurich, Switzerland, 2013). (a) An image from a catadioptric system composed of a hyperbolic mirror and a camera. (b) An image from a system composed of a camera and a fisheye lens.

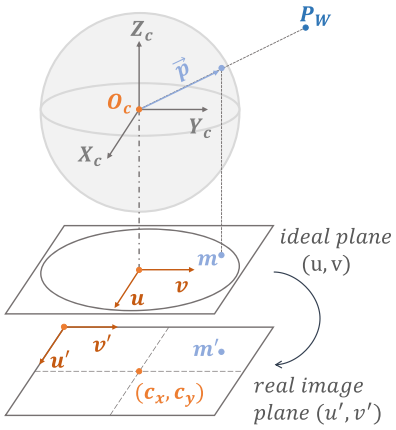


Fig. 2. A scene point P_W is projected onto the unit sphere surface. This way, the 3D unit vector $\vec{p}$ is obtained. Then, it is mapped to a point $m = [u, v]$ on the ideal plane through a function. This ideal plane point is transformed to a point $m' = [u', v']$ in the real image plane (pixel coordinates) by an affine transformation.

3. Catadioptric and fisheye vision systems

In this work, two different visions systems are used: one catadioptric system (omnidirectional camera) and one camera with a fisheye lens. Fig. 1 shows two images of the same scene captured by each of these systems.

The mathematical model of a catadioptric or fisheye camera is more complex than a standard perspective camera. The lens causes refraction, and the mirror produces reflection, so the model should take these effects into account. There are many works in the literature to estimate the model of an omnidirectional camera. The first unified model for central catadioptric systems, that is, cameras using a parabolic, hyperbolic, or elliptical mirror, was proposed by Geyer and Daniilidis (2000). They determine that this type of camera can be modelled by a projection of the 3D scene point onto a unit sphere centred in the effective viewpoint, followed by a perspective projection onto a plane. This model was developed specifically for central catadioptric cameras, so it is not valid for fisheye cameras. Ying and Hu (2004) presented an extension of this model that can be used to model fisheye cameras as well. With respect to Scaramuzza et al. (2006a) all central catadioptric cameras can be represented through an exact parametric function. Still, the projective model depends on the lens field-of-view and varies from camera to camera in the case of the fisheye lenses. Therefore, the approximation of Ying and Hu (2004) for a fisheye camera through a catadioptric one, only works with limited accuracy (Siegwart et al., 2011). To overcome this problem, Scaramuzza et al. (2006a, 2006b) proposed a new unified model for catadioptric and fisheye cameras. In this case, the authors use a Taylor polynomial, whose coefficients and degree are found through a calibration phase.

As mentioned at the beginning of this section, we use catadioptric and fisheye images, so we use this model due to its suitability for both types of cameras (Scaramuzza et al. 2006a, 2006b). Fig. 2 shows the projection following this unified model proposed in (Scaramuzza et al. 2006a, 2006b). A scene point P_W, expressed in the world reference frame can be expressed in the fisheye/mirror reference frame P_C by using the extrinsic parameters. This 3D point is projected onto the unit sphere surface obtaining the unit vector $\vec{p}$ emanating from the centre of the reference frame O_c. Then, the pixel point m is obtained through an imaging function g (see Eq. (1)) and an affine transformation (Scaramuzza et al. 2006a, 2006b).

$$\lambda \cdot g(m) = \lambda \begin{bmatrix} u \\ v \\ f(u,v) \end{bmatrix} = P_c = [\mathbf{R}|\vec{t}]P_w \tag{1}$$

4. Relative pose estimation

Estimating the relative pose between two images taken from different positions is a crucial problem in visual navigation. This technique is known as Visual Odometry (Fraundorfer and Scaramuzza, 2012; Scaramuzza and Fraundorfer, 2011).

To solve the feature-based visual odometry problem, the algorithm can be principally decomposed into three different blocks: (1) feature detection and description, (2) feature matching (or tracking), and (3) motion estimation. The first step consists in identifying points of interest in the image and representing the region around each one as a compact vector, named descriptor, which is used to compare features in different images. The second step consists in detecting the pixel points corresponding to the same 3D point in the pair of images (i.e. finding the matches). Finally, the third step consists in estimating the relative camera motion between the pair of images taken at different times (Scaramuzza and Fraundorfer, 2011). Depending on the dimension of the feature correspondences, there are three techniques to carry out this last step (Yousif et al., 2015): motion estimation from 3D feature correspondences (3D to 3D); from 3D feature and 2D image feature correspondences (3D to 2D); and from 2D image feature correspondences (2D to 2D). In the last method, both feature correspondences are specified in 2D image coordinates, so the relative motion is recovered by the epipolar geometry (see Fig. 3), concretely by the essential matrix **E**. In this work, Standard Method (SM) refers to an algorithm composed only of the three blocks mentioned at the beginning of the previous paragraph, where the technique employed in the motion estimation block is the epipolar geometry (2D to 2D).

The essential matrix depends only on the camera motion parameters that can be recovered only up to a scale factor. This matrix encodes the relative motion parameters between a pair of images and, in consequence, can be defined as:

$$\mathbf{E} = [t]_x \mathbf{R} \tag{2}$$

where $\mathbf{R}$ is the rotation matrix and $[t]_x$ is the skew-symmetric matrix of the translation vector $\vec{t} = [t_x, t_y, t_z]$. After following the process described in Hartley and Zisserman (2003), the relative pose is recovered.

The relative pose can be expressed using angular parameters. Firstly, the coordinates (t_x, t_y, t_z) of the translation vector can be transformed into spherical coordinates. In other words, the relative position between the two camera poses is determined by a radial distance ρ (from the centre of the camera frame at the first pose to the camera centre at the second pose), an elevation angle β and an azimuth angle ϕ.

$$\phi = atan2(t_y, t_x) \tag{3}$$

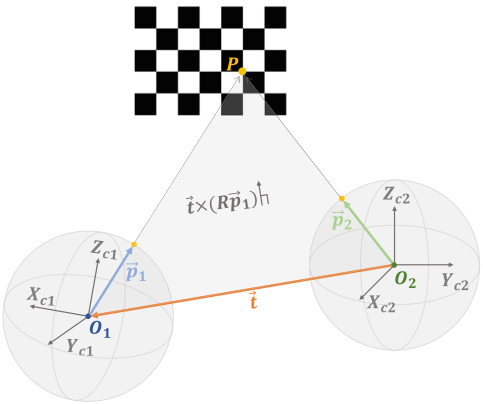


Fig. 3. Epipolar geometry of cameras with wide field of view. A scene point P is projected on the sphere surface of the first and second camera as $\vec{p}_1$ and $\vec{p}_2$, respectively. A rotation matrix and translation vector relate both camera reference systems. Therefore, $\vec{t}$, $\mathbf{R}\vec{p}_1$ and $\vec{p}_2$ lie in the epipolar plane (coplanarity condition).

$$\beta = atan2(t_z, \sqrt{t_x^2 + t_y^2}) \tag{4}$$

$$\rho = \sqrt{t_x^2 + t_y^2 + t_z^2} \tag{5}$$

Secondly, the orientation **R** can be defined by using the Euler angles: yaw (rotation θ around the Z-axis), pitch (rotation γ around the Y-axis) and roll (rotation α around the X-axis). In short, the relative pose is given by six parameters, which can be seen in Fig. 4, five of them are angles (θ, γ, α, ϕ, β) and the remaining one is a scale factor (ρ). This work focuses on the estimation of the angular parameters.

5. Relative pose estimation based on the vehicle model

Some steps of this method require that the relation between the camera frame and the world frame is well-known since the objective is to obtain a 3D model of the environment. Consequently, the mapping from pixel to world coordinates, and vice-versa, will be carried out. In this work, we try to solve the visual odometry problem for a mobile robot that navigates without knowing its following pose, therefore, the camera pose is not known. Nevertheless, assuming that the camera is on-board of a mobile robot, then an approximation of the next camera pose can be obtained by using the probabilistic odometry motion model presented by Thrun et al. (2005). Since the ground truth is available, these data can be modelled as odometry data (by adding some amount of noise), and, after that, the next pose can be estimated. The mobile robot moves from t to $t + 1$, and then the image I_{t+1} is captured. It means that the odometry information, which is usually provided by wheel sensors, is available when the image is processed. Therefore, the odometry-based motion model can be used as an estimation of the relative pose, but it is only used to map the 3D model and image points.

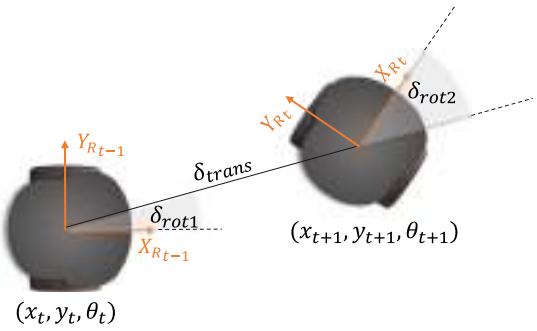


Fig. 5. Parameters of the odometry-based motion: first rotation δ_{rot1}, translation δ_{trans} and second rotation δ_{rot2}.

5.1. Odometry motion model

For a planar environment, the mobile robot state $\vec{x}$ is represented by a point (x, y) and a rotation angle θ that determines the orientation. The odometry-based motion model describes the movement of a mobile robot between two consecutive poses (from $\vec{x}_t = (x_t, y_t, \theta_t)$ to $\vec{x}_{t+1} = (x_{t+1}, y_{t+1}, \theta_{t+1})$) as a sequence of three steps: an initial rotation δ_{rot1}, followed by a straight line motion (translation) δ_{trans} and final rotation δ_{rot2} as illustrated in Fig. 5.

After obtaining $\vec{x}_t$ and $\vec{x}_{t+1}$ from the ground truth data, the parameters of the odometer model can be computed as:

$$\delta_{rot1} = atan2(y_{t+1} - y_t, x_{t+1} - x_t) - \theta_t \tag{6}$$

$$\delta_{trans} = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \tag{7}$$

$$\delta_{rot2} = \theta_{t+1} - \theta_t - \delta_{rot1} \tag{8}$$

In the ideal case, these values would be the same as the ones obtained using the odometer readings, but it does not happen in a real operation. In that case, the measurements provided by the odometer are given by the true motion with independent noises for each one of these motion parameters. The noise is modelled as a zero-mean Gaussian distribution with variance σ and it is denoted as $\epsilon(\sigma)$. Then, the measured parameters are:

$$\hat{\delta}_{rot1} = \delta_{rot1} + \epsilon(\alpha_1 \delta_{rot1} + \alpha_2 \delta_{trans}) \tag{9}$$

$$\hat{\delta}_{trans} = \delta_{trans} + \epsilon(\alpha_3 \delta_{trans} + \alpha_4(\delta_{rot1} + \delta_{rot2})) \tag{10}$$

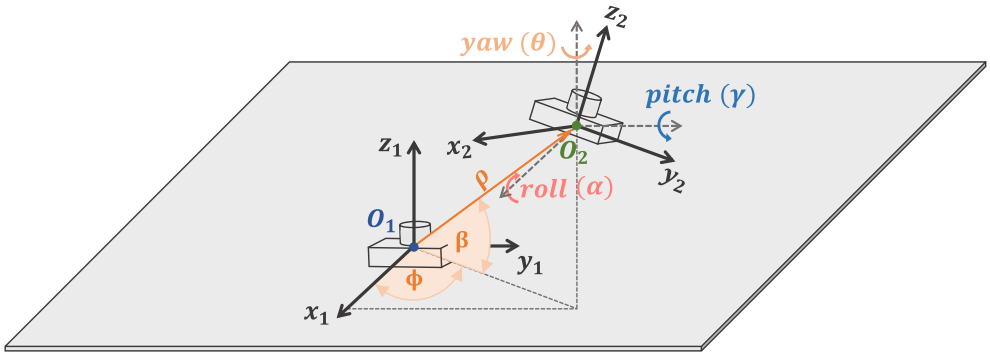


Fig. 4. The relative pose recovered from **E** can be described by six parameters. The orientation can be defined as three successive rotations: around the Z-axis $\mathbf{R}(\theta)$, Y-axis $\mathbf{R}(\gamma)$ and X-axis $\mathbf{R}(\alpha)$. The position is given by two angles (β and ϕ) and a scale factor (ρ).

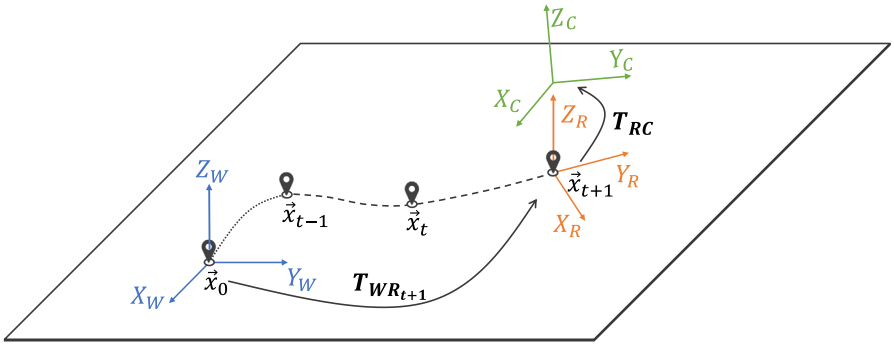


Fig. 6. Coordinate Systems: world frame, mobile robot frame and camera frame.

$$\hat{\delta}_{rot2} = \delta_{rot2} + \epsilon(\alpha_1 \delta_{rot2} + \alpha_2 \delta_{trans}) \tag{11}$$

where the α_1, α_2, α_3 and α_4 parameters model the noise caused by drifts and slipping (translation and rotation). Finally, the initial odometry can be calculated as:

$$\hat{x}_{t+1} = x_t + \hat{\delta}_{trans} \cos(\theta_t + \hat{\delta}_{rot1}) \tag{12}$$

$$\hat{y}_{t+1} = y_t + \hat{\delta}_{trans} \sin(\theta_t + \hat{\delta}_{rot1}) \tag{13}$$

$$\hat{\theta}_{t+1} = \theta_t + \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \tag{14}$$

The odometry is a relative positioning technique (Aqel et al., 2016), so there is no fixed mapping between the coordinates used by the robot's internal odometry and the world coordinates. To solve it, the world reference system has been fixed in the initial state of the mobile robot $\vec{x}_0$, then the relative pose of the mobile robot at the instant $t + 1$ with respect to the world frame $\mathbf{T}_{\mathbf{WR_{t+1}}}$ is given by a rotation matrix around the Z-axis $\mathbf{R_z}(\hat{\theta}_{\mathbf{t+1}})$ and a translation in the XY plane $\vec{t} = (\hat{x}_{t+1}, \hat{y}_{t+1}, 0)$:

$$\mathbf{T}_{\mathbf{WR_{t+1}}} = \begin{bmatrix} \cos\hat{\theta}_{t+1} & -\sin\hat{\theta}_{t+1} & 0 & \hat{x}_{t+1} \\ \sin\hat{\theta}_{t+1} & \cos\hat{\theta}_{t+1} & 0 & \hat{y}_{t+1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

Consequently, the relationship between the camera and world coordinate system $\mathbf{T}_{\mathbf{WC_{t+1}}}$ can be computed assuming that the position of the camera with respect to the mobile robot $\mathbf{T}_{\mathbf{RC}}$ is fixed and well-known.

$$\mathbf{T}_{\mathbf{WC_{t+1}}} = \mathbf{T}_{\mathbf{WR_{t+1}}} \cdot \mathbf{T}_{\mathbf{RC}} \tag{16}$$

In other words, $\mathbf{T}_{\mathbf{WC_{t+1}}}$ is the matrix that transforms the points from the camera frame into the world frame. Fig. 6 shows these reference systems.

6. Adaptive probability-oriented feature matching (APOFM)

This section synthesizes the basis of the APOFM (Valiente et al., 2018) and the improvements proposed in the present paper to improve its performance. In each iteration, the corresponding points (matches) between the images are obtained. A pair of feature points (m_1 and m_2) are considered matched points if their feature descriptors are similar. Therefore, this means that these feature points are the projection of the same 3D scene point. Consequently, if this point appears projected on the next images, and providing it continues to be considered a matching in other iterations, it presents a high probability in the model. Hence, the associated probability with each point is updated at every iteration. Fig. 7 shows the block diagram with the most representative steps of this process.

The model is obtained by using the GP (Rasmussen and Williams, 2006) that is defined as a collection of random variables, a finite number of which have a joint Gaussian distribution, whose input is a set of 3D points (Section 6.2). Hence, it is necessary to recover, previously, the 3D coordinates of each pair of correspondences (Section 6.1). This problem is known as triangulation (Hartley and Sturm, 1997). After that, the environment model is updated using the matches between the previous and current image, and the pose of the next image with respect to the current one is calculated.

The problem of visual odometry is solved following the algorithm described in Section 4. However, some steps have been added and modified in order to improve the matching search. For instance, some new steps have been inserted between the feature detection and feature matching search. Now the search of the feature matchings is not performed with all the feature points detected in the image corresponding to the next pose. The search is only focused on these points considered as candidates. In broad lines, after detecting the feature points in the image taken at the next pose I_{t+1}, using SURF (Bay et al., 2006), the coordinates of the output of the GP are expressed into the frame of the camera at the next time instant $t + 1$. To do it, we use the transformation between world and camera frame calculated in Section 5.1 by the odometry motion model. Next, these points are projected on the image using the calibration parameters (Section 6.3). The next step consists in determining how many detected SURF points are candidates, based on their proximity to a projected probability point (Section 6.4). After that, the search for matches can be carried out (Section 6.5).

In the first iteration, that is, to estimate the relative pose using the images I_0 and I_1, the method employed is SM since there is no information about matching features, that is, all SURF points of I_1 have the same probability of finding a correspondence in I_0. After that, the triangulation problem is solved with the matched features of this iteration, and these 3D points are the input to the GP. This way, the scene model is available from the second iteration, and the proposed model can be employed from then on.

6.1. Triangulation and false positive record

As already mentioned, the triangulation problem essentially consists in calculating the position of a point in the space, given its projection on at least two views, and the calibration parameters and pose estimation. The basic method to solve this problem is to find the intersection of the lines of sight whose origins are the camera centres (O_1 and O_2) and their direction vectors are given by the projections of the image points on the unit surface sphere ($\vec{p}_1$ and $\vec{p}_2$). To recover the coordinates of the 3D point in the world frame, the centres of the camera and the direction vectors must be expressed in the world reference system. The necessary information to do it can be extracted from the estimated transformation matrix that has been calculated using Eq. (16).

However, the rays may not intersect in the 3D space as a consequence of the presence of noise in the matching of image points. This noise can be produced by lens distortion or errors in the calibration parameters. The first one affects the 3D to 2D mapping, whereas the

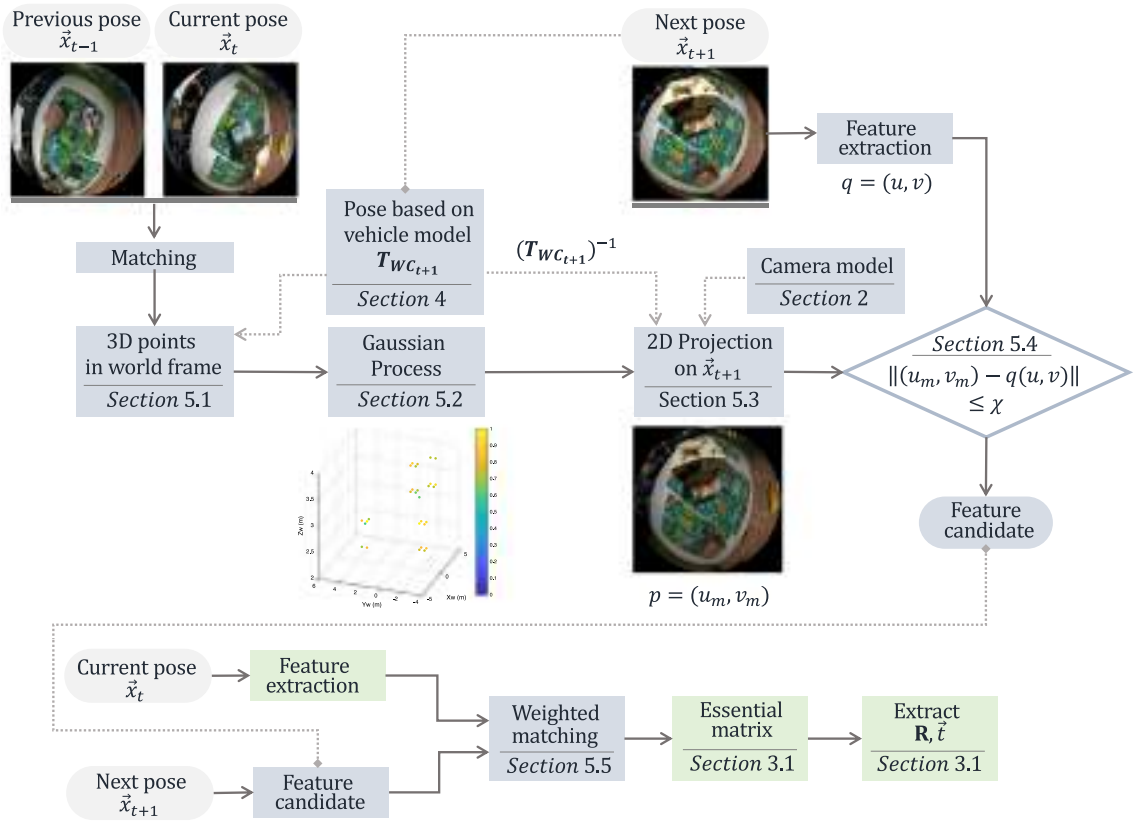


Fig. 7. Block diagram that shows the main parts of the algorithm to create a model of the environment, detailing the sections of the paper in which each part of the algorithm is presented.

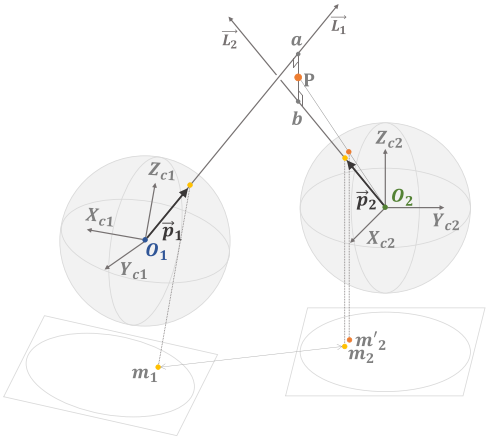


Fig. 8. Triangulation problem: the mid-point approach to recover the 3D coordinates of a point using its projection on a pair of images.

second one makes the 2D to 3D mapping be not precise since the camera model is used in this step. Moreover, some noise may appear due to image processing, such as interest points detection error or the presence of outliers in the correspondence detection. As a consequence, the solution to the triangulation problem becomes nontrivial. In the literature, there are different methods to find the best solution; some of them are described in Nair and Nair (2020). In this work, we adopt the mid-point method proposed in Beardsley et al. (1994). Thus the 3D scene point, is approximated by the midpoint of the segment which is perpendicular to both rays with the shortest distance.

Fig. 8 shows that for the first camera there is a ray defined by the origin O_1 and the direction vector $\vec{p}_1$, so its corresponding equation is $\vec{L_1} = O_1 + \lambda_1 \cdot \vec{p}_1$. Similarly, there is a ray equation, whose equation is $\vec{L_2} = O_2 + \lambda_2 \cdot \vec{p}_2$, for the second camera. The first step for the computation of the intersection point is to obtain the points $\vec{a}$ and $\vec{b}$. These points are the intersection of the common perpendicular $\vec{ab}$ with the line $\vec{L_1}$ and $\vec{L_2}$ respectively. In other words, the point $\vec{a}$ satisfies the equation of the first ray, so $\vec{a} = O_1 + \lambda_1 \cdot \vec{p}_1$, and the point $\vec{b}$ satisfies the equation of the second ray, hence $\vec{b} = O_2 + \lambda_2 \cdot \vec{p}_2$.

Due to the fact that the segment ab is perpendicular to both rays, the dot product of its direction vector, and the corresponding of each ray is equal to zero.

$$(\vec{b} - \vec{a}) \cdot \vec{p}_1 = (O_2 - O_1) \cdot \vec{p}_1 + \lambda_2 \cdot \vec{p}_2 \cdot \vec{p}_1 - \lambda_1 \cdot \vec{p}_1 \cdot \vec{p}_1 = 0 \tag{17}$$

$$(\vec{b} - \vec{a}) \cdot \vec{p}_2 = (O_2 - O_1) \cdot \vec{p}_2 + \lambda_2 \cdot \vec{p}_2 \cdot \vec{p}_2 - \lambda_1 \cdot \vec{p}_1 \cdot \vec{p}_2 = 0 \tag{18}$$

After solving the equation system, the unknowns λ_1 and λ_2 are obtained. The intersection point is the average of the points $\vec{a}$ and $\vec{b}$.

$$P = \frac{\vec{a} + \vec{b}}{2} = \frac{(O_1 + \lambda_1 \cdot p_1) + (O_2 + \lambda_2 \cdot p_2)}{2} \tag{19}$$

As mentioned above, sometimes the set of corresponding points may contain wrong matches, named false positives. Fig. 9 depicts this problem. The detected feature point in the first image m_1 and the detected feature point in the second image m_2 are considered as a pair of corresponding features during the matching search. Then, the triangulation problem is solved, and the result is the 3D point P. If P is re-projected on the second image m'_2, it can be observed that its projection is not near the detected feature point in the second image m_2. This means that the feature points are not the projection of the same 3D point (false positive). As a matter of fact, this can be noticed in this figure, where the true 3D point of each feature point is shown. To extend this example, Fig. 10 shows two pairs of matched features where one is a false positive and the other is a true positive.

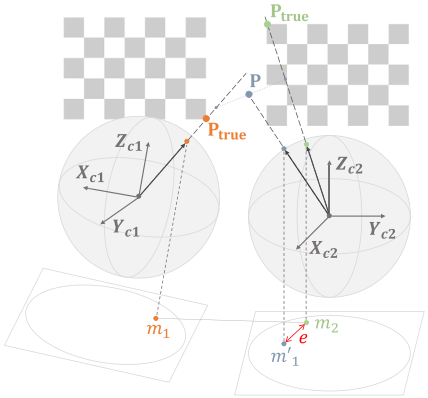


Fig. 9. The detected feature point in the first image • and the detected feature point in the second image • have been considered as a pair of corresponding features. However, it can be observed that the feature points are not the projection of the same 3D point so it is a false positive. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

It is important to note that this contribution aims to serve as a tool for quantifying false positive ratios, and thus evaluating the efficacy of the probability-based matching.

6.2. GaussIan process

Once the 3D coordinates corresponding to each matching feature have been recovered, the next step is to create the probability model with them using the GP.

A GP can be seen as a generalization of the Gaussian probability distribution to function spaces. It means that a probability distribution describes random variables, whereas a GP is a distribution over functions $f(x)$. Therefore, if a Gaussian distribution is given by its mean and covariance, then a GP is formed by a mean function $f_m(x)$ and covariance function $k(x, x')$. So, the GP can be written as:

$$f(x) \sim \mathcal{GP}(f_m(x), k(x, x')) \tag{20}$$

where $x \in \mathbb{R}^d$ and $x' \in \mathbb{R}^d$, are the training and test (query) input points respectively.

The algorithm used to obtain the probabilistic model of the environment is the one proposed by Ghaffari et al. (2018). They developed a technique for occupancy mapping using the GP. As presented in Fig. 11, the algorithm is composed of three main modules: (1) GP regression (Section 6.2.1). (2) Logistic regression classifier that squashes the output of the prior module into probabilities (Section 6.2.2) and leads to the local map. (3) Bayesian Committee Machine (BCM) (Tresp, 2000), which updates the global map incrementally. The output of this algorithm is a probability distribution which is shown in Fig. 12(a) and the modules that compose it are described in depth in the following subsections.

6.2.1. GaussIan process regression

Given a set of n training input points $X = \{x_1, x_2, \dots, x_n | x_i \in \mathbb{R}^3\}$, their corresponding output values arranged as a vector $y = \{y_1, y_2, \dots, y_n | y_i \in \mathbb{R}\}$ and a set of n_t test points $X_* = \{x_{*1}, x_{*2}, \dots, x_{*n_t} | x_i \in \mathbb{R}^3\}$. The mean and covariance of the predictive conditional distribution for test data $f_* | X, y, X_* \sim \mathcal{N}(\bar{f}_*, cov(f_*))$ can be computed as follows:

$$\bar{f}_* = K(X, X_*)^T (K(X, X) + \sigma_n^2 I)^{-1} y \tag{21}$$

$$cov(f_*) = K(X_*, X_*) - K(X, X_*)^T (K(X, X) + \sigma_n^2 I)^{-1} K(X, X_*) \tag{22}$$

where σ_n^2 is the variance of the observation noise and $\mathbf{K}(\cdot, \cdot)$ denotes the covariance matrix of the variables $(\cdot, \cdot)$, for instance, $\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$ is the $n \times n_t$ matrix of the covariances evaluated at all pairs of training X and test points X_*.

In this work, the training input data X are the 3D points P obtained after solving the triangulation problem for each pair of feature correspondences. The target value assigned to each training input point is one ($y_i = 1$) indicating that the projections of this point on the images at $t-1$ and t have been considered as a pair of matched features. Therefore, the training output data y is a vector of ones $y = \{1, 1, \dots, 1\}$. Finally, the test data are the set of spatial coordinates to build the map on. In other words, the motion space of the mobile robot with existing points is evaluated. This map consists of a three-dimensional grid represented by the vectors $X_m = \{x_1 : i : x_{n_x}\}$, $Y_m = \{y_1 : i : y_{n_y}\}$, and $Z_m = \{z_1 : i : z_{n_z}\}$ that are defined by a starting and ending value, and an increment i between their elements, which is denominated the step of the grid ($\Delta grid$). The number of test points is given by the length of these vectors so $n_t = n_x \cdot n_y \cdot n_z$.

6.2.2. Logistic regression classifier

Since the goal is to obtain a probabilistic representation of the environment, the output of the GP regression, that is the prediction (μ_*, σ_*^2) at a test point x_*, must be squashed into the range $[0, 1]$. Hence, a logistic function is used.

$$p(y_* = 1 | X, y) = \frac{1}{1 + exp(-\gamma \omega_i)} \tag{23}$$

where $\omega_i = \mu_{*i} \lambda^{1/2}$ is the weighted mean, $\lambda = \sigma_{min}^2 / \sigma_{*i}^2$ denotes the bounded information associated to each location, σ_{min} is the minimum predicted variance by the GP regression and γ is a positive constant parameter to control the sigmoid shape.

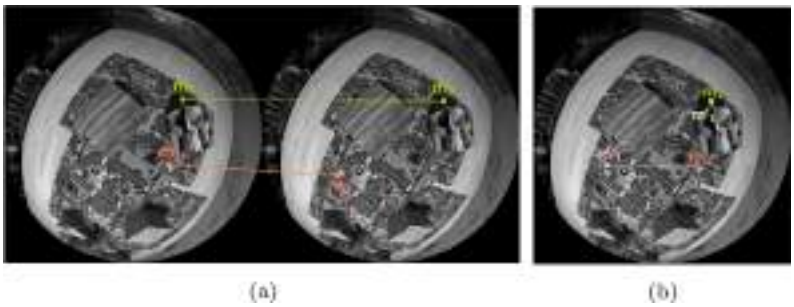


Fig. 10. Detection of false positive: (a) Two pairs of features matches can be seen in this figure. Each feature is symbolized by m_i where the subscript i indicates to which image it belongs. Furthermore, each pair of matched features is represented by a different colour. (b) The calculated 3D points are projected on the first image m'_1 after solving the triangulation problem for each pair of correspondences shown in (a). It can be observed that the projected point m'_1 is nearby the feature point in the case of the green correspondence. Nevertheless, it does not happen the same for the orange matching since this is a false positive. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

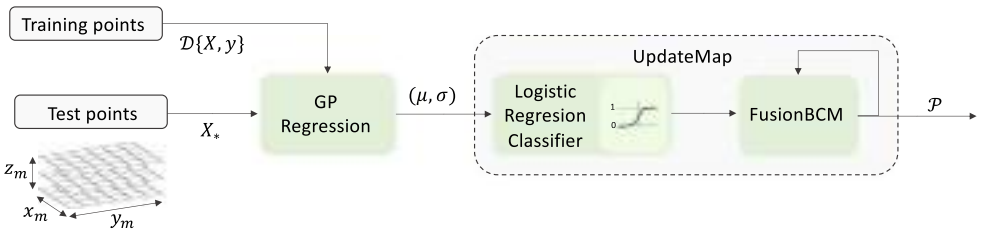


Fig. 11. Block diagram of the algorithm that calculates the probabilistic model of the environment using GP.

6.3. Projection of the model points on the 2D image

A 3D probability distribution has been obtained in the previous section. Given that it determines the probability that the projection of a 3D scene point is a feature correspondence, the output of the GP must be projected on the image at the next pose. In this manner, relevant areas are obtained over the image. If a feature point is detected in one of these areas, then it will probably be a matching feature.

The first step is to express each 3D point of the probability distribution ${}_W P = \left\{ {}_W p_1, {}_W p_2, \dots, {}_W p_n | {}_W p_i \in \mathbf{R}^3 \right\}$ in the camera coordinate system:

$$ {}_{C_{t+1}} p_i = \mathbf{T}_{\mathbf{C_{t+1}W}} \cdot {}_W p_i \tag{24} $$

where $\mathbf{T}_{\mathbf{C_{t+1}W}}$ is the matrix that transforms the points from world to camera frame at $t + 1$. To move the GP output to the next frame pose $t + 1$, an estimation of the relationship between the world and camera frames must be available. This estimation es calculated from the vehicle model (Section 5). In this case, we obtain the matrix that transforms the points from the camera to the world frame $\mathbf{T}_{\mathbf{WC_{t+1}}}$ using Eq. (16). Therefore, taking this into account, the previous equation can be written as:

$$ \begin{aligned} {}_{C_{t+1}} p_i = \mathbf{T}_{C_{t+1}W} \cdot {}_W p_i = \mathbf{T}^{-1}_{\mathbf{WC_{t+1}}} \cdot {}_W p_i = \\ \left[\mathbf{R}^T_{\mathbf{C_{t+1}W}} \quad -\mathbf{R}^T_{\mathbf{C_{t+1}W}} \cdot \vec{t}_{C_{t+1}W} \right] \cdot {}_W p_i \end{aligned} \tag{25} $$

where $\mathbf{R}_{\mathbf{C_{t+1}}W}$ is the rotation matrix that describes the orientation of the camera frame with respect to the world frame and $\vec{t}_{C_{t+1}W}$ is the distance vector from world to camera expressed in world frame. The next step is to calculate the pixel coordinates of each point using the camera model Eq. (1).

Fig. 12(a) shows the 3D probability distribution of feature existence expressed in the world frame. Fig. 12(b) shows these same points with their associated probability in the image at $t + 1$ after performing the transformation between frames and mapping the 3D points in the camera frame to 2D image points.

6.4. Determining candidate features

The last step of this method is to determine which of the detected feature points will be considered as a possible matching candidate from the probability points projected.

The feature points will be candidates if they are near projected points with an associated probability, besides they will be assigned the probability of the nearest point. To carry out this, the Nearest Neighbour (Cover and Hart, 1967) method has been used, which calculates the distances between the test data and each of the training data in order to identify the nearest neighbour.

Given a set of training data $p_1, p_2, \dots, p_n$, and a distance function d, the nearest neighbour search permits finding the closest point in the training dataset to each query point q according to Eq. (26). In the APOFM, the training points are the projected points with an associated probability, whereas the set of query points are the feature points detected.

$$ NN(q) = \arg\min_{p_i} d(p_i, q) \tag{26} $$

There are several types of distance functions which have been used in the literature (Chomboon et al., 2015), such as Euclidean, Mahalanobis, Manhattan, Minkowsky, City-block, and Chebyshev. In this paper, two of these distance metrics have been employed. In the first place, the Mahalanobis distance, whose search of the nearest neighbour has been carried out using the exhaustive method. This search method finds the distance from each detected feature point to all n projected points with an associated probability. In the second place, the City-block distance has been employed to find the nearest neighbour using the Kd-tree algorithm (Bentley, 1975). Finally, each feature point is classified using the distance between itself and its nearest neighbour. Then, a specific threshold χ is imposed on the maximum distance for a feature point to be considered as a candidate. The value of χ is given by the chi-square inverse cumulative distribution function, with n_{dof} degrees of freedom, evaluated at a probability value. In this work, n_{dof} is equal to 2 since this is the dimension of the image points. The probability value is chosen as the one that provides the best results, according to the experiments performed in Section 7.1.

In summary, a feature point is classified as a candidate only if $d(p_i, q) < \chi$. On the contrary, the feature points which do not satisfy this requirement are classified as not candidates and are not taken into account in the next step.

Fig. 12(c) shows the projected points with an associated probability and the detected SURF feature points in the image taken at $t + 1$. After solving the classification problem, the detected SURF feature points which are classified as candidates are represented in Fig. 12(d), with a specific colour based on its probability.

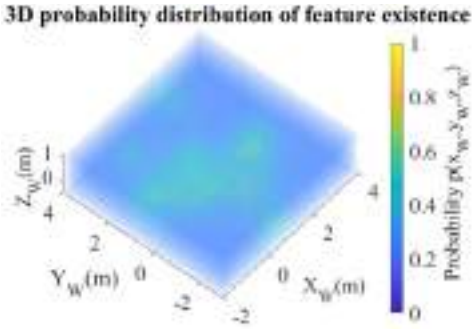
6.5. Image matching

This step consists in searching for similar features between a pair of images, that is, two-dimensional features that are the re-projection of the same 3D point across two different frames. A common approach to this task is to compare all feature descriptors in the first image to all other feature descriptors in the second image. After comparing all feature descriptors using a similarity measure, the correspondence of a feature is established by finding the nearest neighbour in the descriptor space.

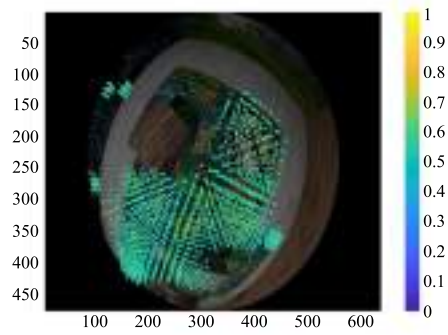
The problem of image matching can be formulated as follows (Hassaballah et al., 2016): after finding a set of interest points and extracting the feature descriptors around each one as a vector of length M, suppose that q_1^1 is one of these points in the first image I_1, and $\mathbf{F_1^1} = [f_1^1(1), f_1^1(2) \dots f_1^1(M)]$ is its feature descriptor. The aim is to find the best matching point q_2^j from the set of N feature points detected in the second image I_2 so, $j = 1, 2 \dots N$. To this end, the feature vector $\mathbf{F_1^1}$ is compared with each keypoint descriptor extracted $\mathbf{F^{j_2}} = [f_2^j(1), f_2^j(2) \dots f_2^j(M)]$ from I_2 by means of a distance function such as the Euclidean.

$$ d_j(\mathbf{F_1^1}, \mathbf{F^{j_2}}) = \left\| \mathbf{F_1^1} - \mathbf{F^{j_2}} \right\| = \sqrt{\sum_{i=1}^{M} (f_2^j(i) - f_1^1(i))^2} \tag{27} $$

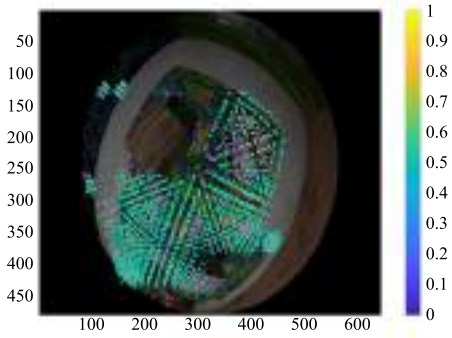
where $j = 1, 2 \dots N$ and N is the number of keypoints in I_2. Once all the distances are calculated, the nearest neighbour is searched, that is, the one with the minimum distance d_{1st}. The feature point associated is



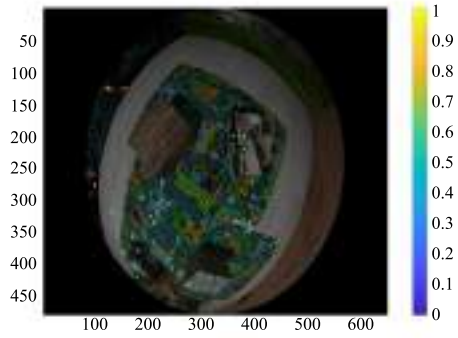
(a) Probability model of the environment. 3D probability distribution of feature existence (Section 6.2)



(b) Projected probability on the image at the next pose I_{t+1} (Section 6.3)



(c) Detected SURF points are plot on the image as squares in magenta colour ■.



(d) Candidate points with an associated probability (Section 6.4).

Fig. 12. The 3D points with a probability (a) are transformed from the world frame to the camera frame at pose $t + 1$. Then, they are projected on the image and their pixel coordinates are obtained (b). Once the projected points and the detected SURF points are expressed in the image at $t + 1$ (c), the process to extract SURF points as candidate is carried out. Finally, we obtain a set of SURF points (candidates) with an associated probability (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

accepted as a correspondence of q_1^1 only if this distance is smaller than a threshold.

However, this requirement is not enough to discard ambiguous matches, and this is due to the fact that some descriptors are much more discriminative than others. For this reason, another condition based on Nearest Neighbour Distance Ratio (NNDR) (Lowe, 2004; Mikolajczyk and Schmid, 2005) has been used to find the best match. This method considers a matching is reliable only if the closest neighbour is significantly closer than the closest incorrect match. Thus, the distance ratio between the nearest $\mathbf{F_2^{1st}}$ and the second nearest $\mathbf{F_2^{2nd}}$ image descriptor is used.

$$NNDR = \frac{d_{1st}}{d_{2nd}} = \frac{\left\|\mathbf{F_1} - \mathbf{F_2^{1st}}\right\|}{\left\|\mathbf{F_1} - \mathbf{F_2^{2nd}}\right\|} \leq th_{ratio} \tag{28}$$

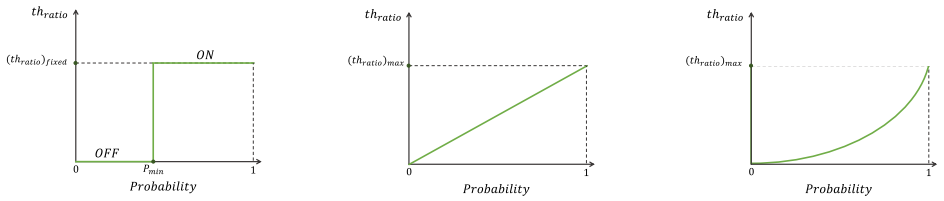
where d_{1st} and d_{2nd} are the Euclidean distances to the nearest and second nearest neighbour respectively. A correct match will have a distance ratio lower than a specific threshold, whereas an ambiguous match or an incorrect match will have a distance ratio close to one (Hassaballah et al., 2019).

Taking all this information into consideration, the feature point associated to the nearest feature descriptor (the one with minimum Euclidean distance) is considered as the best match only if this distance is lower than a matching threshold ($th_{matching}$) and the ratio between the nearest and the second closest match is smaller than a ratio threshold (th_{ratio}).

As presented in the previous section, the APOFM employs the 3D probability distribution to obtain the set of candidate points. In the present paper, we propose using this probability information to weigh the matching search as well. On account of that, the improved APOFM employs a weighted and dynamic matching evaluated under three custom functions, as presented in Fig. 13. The value of the $th_{matching}$ is constant; by contrast, the th_{ratio} value will depend on the function used (step, linear or square). In the weighted matching with a step function, which is shown in Fig. 13(a), only the projected probability points whose associated probability is higher than a threshold (P_{min}) are considered. The points whose probability is lower than P_{min} are not considered in the matching search. In the weighted matching with a linear function, which is shown in Fig. 13(b), all the projected probability points are taken into account and the value of the th_{ratio} is established according to the associated probability and a linear function. In the weighted matching with a square function, which is shown in Fig. 13(c), all the projected probability points are taken into account and the value of the th_{ratio} is established according to the associated probability and this function.

7. Results

In order to have objective evidences of the performance of this work, this section presents results evaluated in a publicly available dataset, with the inclusion of a benchmark of the different methods introduced in Table 1. As stated in the introduction, one of the goals of this work is to compare the performance of the improved APOFM (Section 6) solving the visual odometry with a SM (Hartley and Zisserman, 2003) described in Section 4. In addition to this, given that the main feature of the APOFM is the optimization of the matching search regarding to false positives, we have also compared SM with outlier rejection by means of RANSAC (Scaramuzza, 2011). The code implemented for this purpose is an open-source available in Yan (2011) which has been adapted to estimate the essential matrix. After performing a study, the values of the RANSAC parameters have been optimized to obtain the best estimation of the relative pose. We denote it as



(a) Step. (b) Linear. (c) Square.

Fig. 13. Value of th_{ratio} based on (a) step, (b) linear or (c) square function.

Table 1
Summary of the different methods and variations employed during the experiments.

Identification	Method	Function th_{ratio}	Parameters of the function
SM	Standard method (Hartley and Zisserman, 2003)	–	–
SM+RANSAC	Standard method and RANSAC to remove outliers (Scaramuzza, 2011)	–	–
WM-SF0.6	Improved APOFM	Step (Fig. 13(a))	$(th_{ratio})_{fixed} = 0.4$ and $P_{min} = 0.6$
WM-SF0.7	Improved APOFM	Step (Fig. 13(a))	$(th_{ratio})_{fixed} = 0.4$ and $P_{min} = 0.7$
WM-LF	Improved APOFM	Linear (Fig. 13(b))	$(th_{ratio})_{max} = 0.4$
WM-SqF	Improved APOFM	Square (Fig. 13(c))	$(th_{ratio})_{max} = 0.4$

SM+RANSAC. These comparisons are carried out using images taken with two different types of wide field of view cameras: fisheye and catadioptric.

In this regard, we have used the image dataset available in Robotics and Perception Group, University of Zurich, Switzerland (2013) and Zhang et al. (2016) composed by synthetic images generated with Blender. These images were rendered with two different camera models (fisheye and catadioptric) that were moving along the same trajectory in an indoor pixels for the fisheye model (180° FOV), and another sequence with the same number of images and resolution for the catadioptric model have been obtained.

On this matter, two plots have been obtained for each experiment, one using images captured by the catadioptric camera (they will be on the left side of the figures in the following subsections) and the other with images captured by the fisheye camera (these will be on the right side) and a comparative evaluation is performed. Altogether, six methods are considered which are summarized in Table 1. The first of them is the Standard Method (SM) and the remaining ones are variations of the improved APOFM, denoted in this section as WM (Weighted Matching). Focusing attention on the latter, the changes are related to the feature matching search step (Section 6.5). The second method (WM-SF0.6) considers a Step Function (SF) (Fig. 13(a)) to set th_{ratio}, with $P_{min} = 0.6$. The third method (WM-SF0.7) considers the same function with $P_{min} = 0.7$. The fourth method (WM-LF) uses a Linear Function (LF) to set th_{ratio} (Fig. 13(b)) with $(th_{ratio})_{max} = 0.4$ and, finally, method 5 (WM-SqF) makes use of a Square Function (SqF) (Fig. 13(c)) with $(th_{ratio})_{max} = 0.4$.

7.1. Parameters: $\Delta grid$ and χ

The APOFM depends mainly on two parameters. The first one is the value of χ, as mentioned in Section 6.4. This parameter is the threshold that determines if a detected feature point is a candidate to be a matching feature according to the distance between itself and the nearest projected probability point. The second parameter delimits the number of test points n_t that the GP has to treat.

Thus, the first experiment tries to evaluate the influence of these parameters upon the localization error and computation time. Thereby, the experiment will permit selecting optimum values for both parameters: χ and $\Delta grid$, so that the localization error is small and the computation time is admissible. Given that the third method (WM-SF0.7) is the most restrictive in comparison with the other proposed methods (i.e. fewer feature points are candidates), it has been employed for this experiment.

Therefore, the algorithm will be run for different values of χ and $\Delta grid$ whereas the values of the other parameters are fixed. A range of possible values for χ and for the $\Delta grid$ has been defined. For each possible combination of values of these parameters, the relative pose between each image of the dataset (t) and its three successive ones ($t + 1$, $t + 2$, $t + 3$) has been calculated. Then, the mean value of all localization errors obtained at each iteration is calculated. Fig. 14 shows the translation error (i.e. the error when the azimuth ϕ angle is estimated) which is shown with a specific colour based on its value. It is worth highlighting that, in some cases, it is not possible to estimate the relative pose because the number of correspondence pairs are not enough to obtain the essential matrix. These cases are represented with white colour in Figs. 14 and 15. It usually happens when (a) the value of χ is small and (b) the value of the $\Delta grid$ is high. The first condition denotes that χ is more restrictive in terms of distance and, consequently, the number of feature points considered candidates will be fewer. As a result of the second condition, the probability model of the environment will be represented by a low number of points, and the result is a loss of 3D information.

As Fig. 14 shows, the behaviour is different for each type of camera. In the case of the catadioptric vision system (Fig. 14(a)), the error is smaller when the value of the $\Delta grid$ is low, and the value of χ is high; in other words, when the method is less restrictive (i.e. there are more points to represent the scene and more feature points are considered as candidates). In the case of the fisheye camera (Fig. 14(b)), the smallest error is obtained when both parameters take values in the middle of the range of possible values.

Next, Fig. 15 shows the computation time of the process. In this case, the influence on the calculation time is the same, regardless of the camera type. In both cases, the time is shorter as the $\Delta grid$ is increased. This result makes sense given that the higher the value of the $\Delta grid$, the lower the number of test points is. This means that the number of points that the GP has to treat is lower and, as a consequence, the computation time is also lower. As regards the χ parameter, it can be observed that when it increases, so does the computation time. We could expect this fact since this means that more SURF points are considered as candidates, so both the matching search and the GP (training points) have to process more data in terms of points. However, the increment of the computation time is small compared to the one caused by the $\Delta grid$. Therefore, we can say that the GP has more influence on the computation time than the other parts of this process (e.g. the image processing), and it does not depend on the camera type.

Taking all the above information into account, the values of χ and $\Delta grid$ have been chosen to obtain a good balance between error and time for both kinds of cameras. As for the value of the $\Delta grid$, the

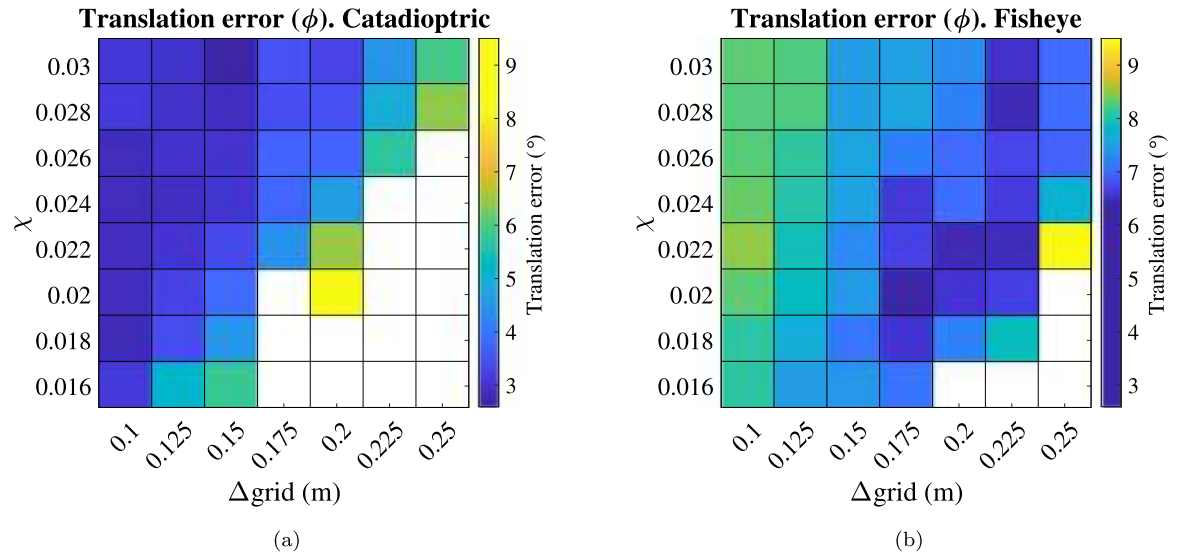


Fig. 14. The influence of the values of the $\Delta grid$ and χ upon the translation error when using either (a) a catadioptric or (b) a fisheye camera.

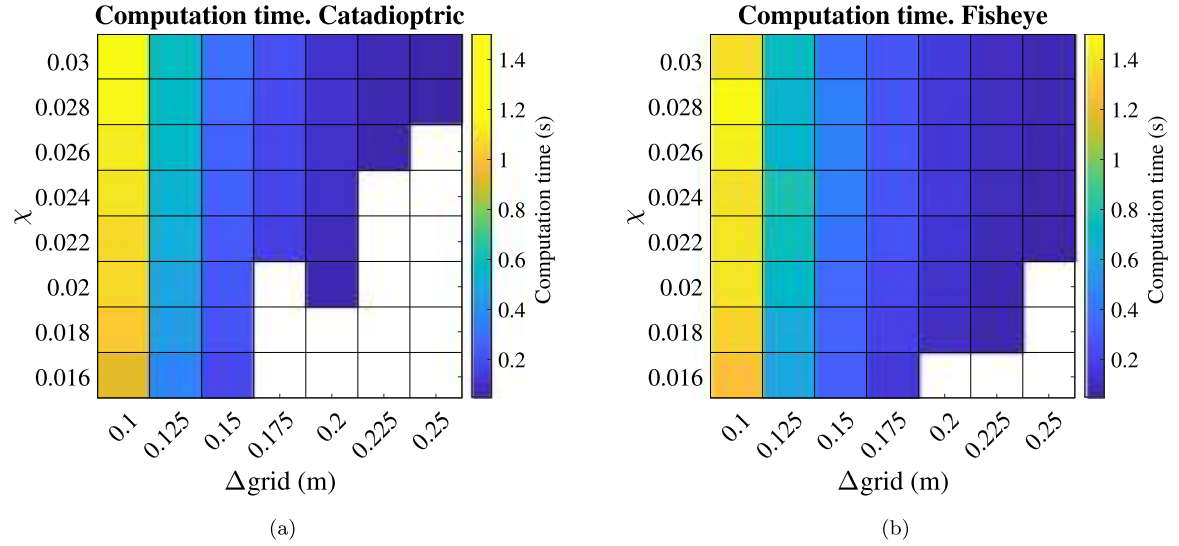


Fig. 15. The influence of the values of the $\Delta grid$ and χ upon the computation time when using either (a) a catadioptric or (b) a fisheye camera.

selected value is 0.15, since the results obtained in both cases show a good balance. While it is true that they are better for the catadioptric camera, higher values could lead to a case in which it is not possible to estimate the pose. Additionally, according to Fig. 15(b), the best result for this $\Delta grid$ occurs when χ is 0.018, so in the following sections, the experiments are carried out with these optimum values.

7.2. Number of feature matches

This subsection studies the number of SURF feature points corresponding to the next image I_{t+1} that have been considered in the search of matching features, and how many of them have found matches in the current image I_t. Fig. 16 shows the number of these sets of points considering images captured in different times, labelled as d_1, d_2, d_3. The first distance specified as d_1 (Fig. 16(a) and (b)) denotes that the algorithm considers the images I_t and I_{t+1}. The second distance specified as d_2 (Fig. 16(c) and (d)) means that the algorithm estimates the relative pose between the images I_t and I_{t+2}. Finally, in the case of d_3, the images taken at t and $t+3$ have been employed (Fig. 16(e) and (f)). In each sub-figure, the columns denote number of points (left axis). The first column corresponds to the SM, so it represents all SURF points detected in the next image I_{t+i} (where $i = 1, 2, 3$) and the number of them which have been found as a match in the current image I_t. In the second column, the same results are represented but employing RANSAC to estimate the essential matrix. The other columns show the results when the improved APOFM with specific variations is employed (Table 1). In these cases, the points considered in the searching of matching features are the candidate points (Section 6.4), therefore the number of these points and how many of them have been found as match are represented in each one of these columns.

Firstly, we analyse the results of the SM with each type of camera. Even though the number of detected SURF points is higher for the fisheye camera, the results show that the number of feature matches is higher for the catadioptric camera. This effect is likely to appear when the field of view is higher. Comparing to SM, the number of matches with SM+RANSAC is lower, as expected, since it removes those matches that do not fit well the model. It leads to lower values of matching ratio, especially with fisheye images.

Secondly, about the APOFM, comparing the number of candidate points, it can be said that more points have been determined as candidates with the fisheye camera than with the catadioptric one. However, if we calculate the ratio between them and the number of matching

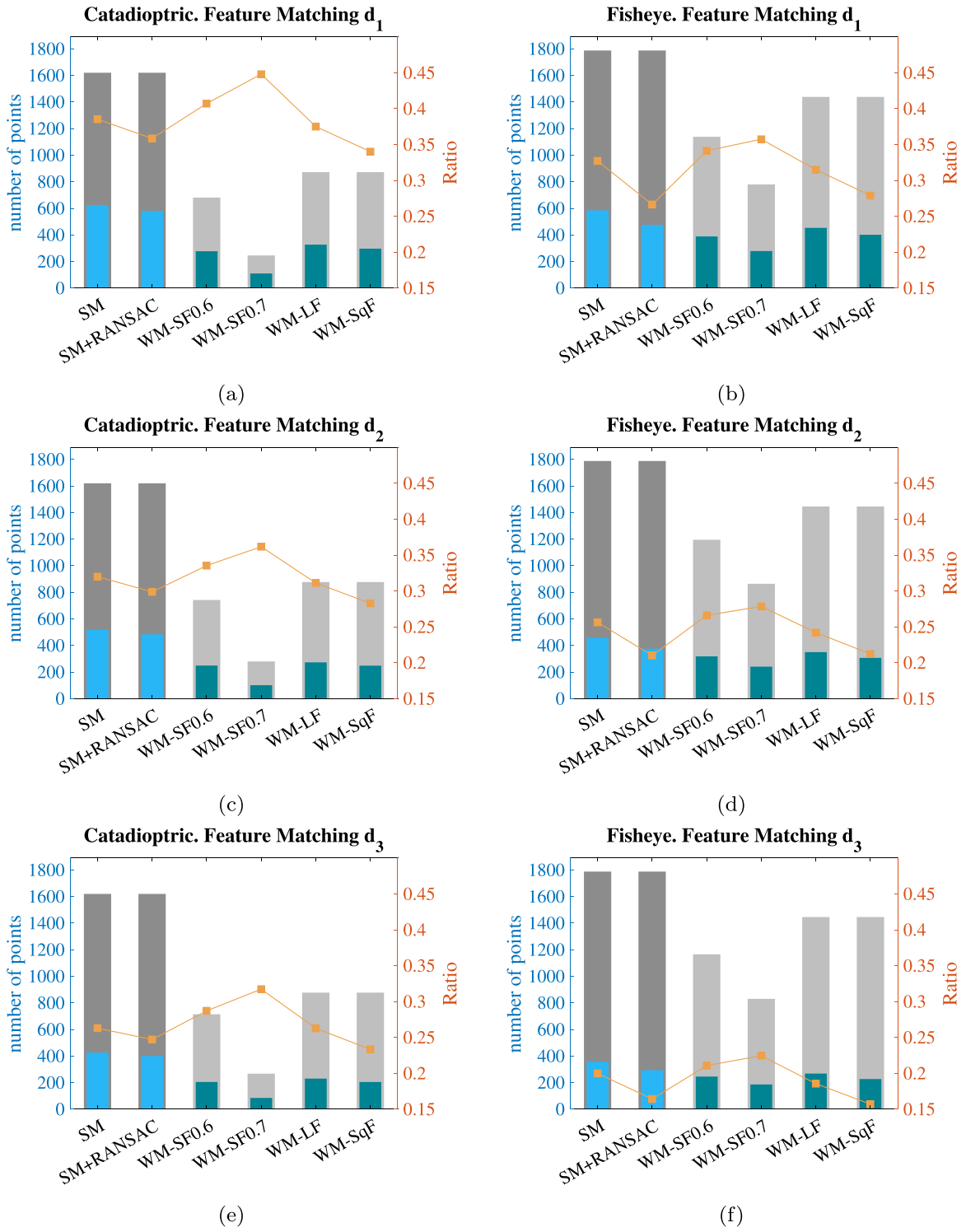
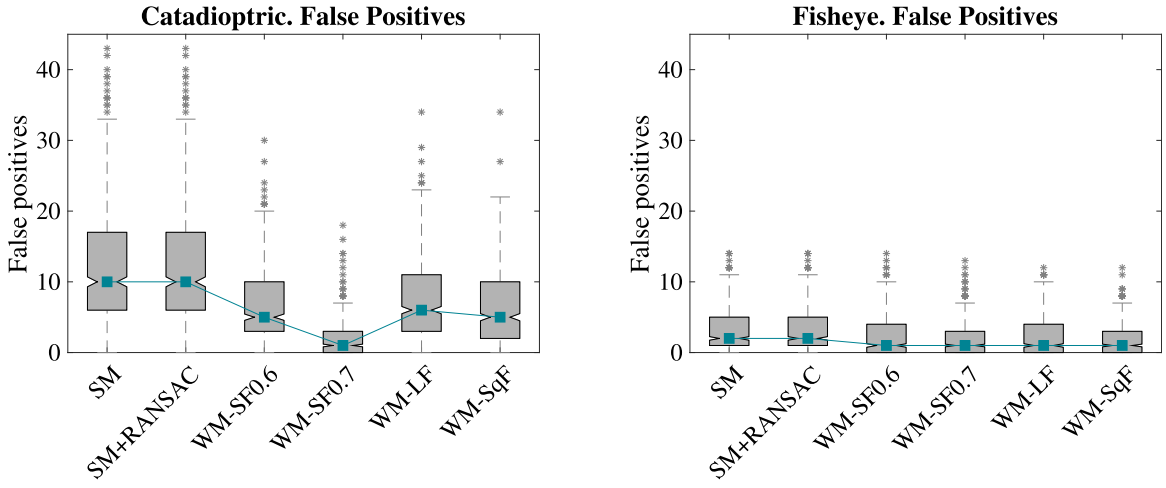


Fig. 16. Number of SURF points and ratio between the number of considered and matched points in the next image I_{i+i}, with (a), (b) $i = 1$; (c), (d) $i = 2$ and (e), (f) $i = 3$. The left axis shows the total number of points (num SURF ■); the number of them that have found a match in the current image I_i using SM (Standard Matching ■); the number of points considered as candidates by the APOFM (num candidates ■) and how many of these latter have found matches (Proposed Matching ■). The right y-axis shows the ratio ■ between the number of feature points used during the matching step and the number of them that have found a match. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

points (this ratio is represented as an orange tendency whose values are shown in the right axis of Fig. 16), the catadioptric camera provides better results. In other words, many candidate points are not found as correspondence in the case of the fisheye camera. This may be due to the fact that these candidates are extracted by metric distance to projected points with an associated probability (pixel frame). However, given the nature of the fisheye images, the reprojected rays of these candidates might be practically coincident with more than one 3D point. As a result, several 3D points might be associated with the same pixel location, thus losing the coincidence of their visual descriptors. Finally, the matching discards these points since it does not find corresponding visual descriptor.



(a) Catadioptric camera (b) Fisheye camera

Fig. 17. Distribution of the number of false positives in the images captured by (a) catadioptric and (b) fisheye camera. The SM and the variations of the improved APOFM (WM-SF0.6, WM-SF0.7, WM-LF and WM-SqF) have been compared.

Considering the number of feature points, the third method WM-SF0.7 provides the best results for both types of cameras since the ratio is high in this case. Even so, it is necessary to study its behaviour regarding the false positives and the localization error before determining if this method is the best one.

To complete this section, we discuss the effect of the distance between the images in the number of matched points (i.e. d_1, d_2, d_3). After observing the results achieved for each distance with the catadioptric camera (Fig. 16(a), (c) and (e)), and with the fisheye camera (Fig. 16(b), (d) and (f)), the conclusion is that, regardless the type of camera, the number of matches is lower when the distance between the images is higher. This difference of matches is lower in the case of the catadioptric camera.

7.3. False positives

A high number of feature matches does not indicate *per se* that a specific method is more effective. It is true that the higher the number of matches, the more information about the relative motion, and consequently, the localization error is expected to be smaller. However, as mentioned in Section 6.5, some of these matches may be false positives and may lead to a wrong estimation of the relative camera pose.

In this sense, Fig. 17 shows the number and distribution of false positives with each method by means of a boxplot. Each one represents all the false positives between the current image I_t and the three successive ones I_{t+1}, I_{t+2} and I_{t+3}. Once the plots have been observed, the conclusion is that the range of the number of false positives is greater using the SM than using the variations proposed in this work. In particular, the third proposal WM-SF0.7 demonstrates a more condensed distribution, fact that implies a lower number of false positives, but also less dispersion.

Even though the results for the SM and SM+RANSAC seem similar, there is a small decrease in the number of false positives. Still, this difference is much more significant with the improved APOFM.

After keeping the result obtained in this subsection as well as the previous ones in mind, it can be said that the feature matches are more robust in all cases in which the improved APOFM has been employed, since the number of false positives is smaller, though this implies that some true positives have also been eliminated. For this reason, it is also necessary to study the localization error, based on the method employed.

Regarding the type of camera, there are fewer false positives in the images taken by a fisheye camera than by a catadioptric camera. Furthermore, the lower whisker of the boxplots does not exist for the improved APOFM since the median is near to zero. In the majority of the iterations to obtain the relative pose, the number of false positives obtained is between zero and a small value.

7.4. Localization error

One of the aims of the work is to solve the localization problem; hence it is necessary to make a study about the error obtained after estimating the relative pose with each of the methods. Two different distance measures have been applied to determine the candidate feature points. As mentioned in Section 6.4, they are the Mahalanobis distance (exhaustive search algorithm) and City-Block distance (kd-tree search algorithm). Figs. 18 and 19 show the angular error made in the estimation of the relative pose by means of a bar graph, where the first bar corresponds to the error using the SM. Each variation of the improved APOFM has two bars. The first one shows the error using the Mahalanobis distance to determine candidate feature points; the second bar represents the angular error when the distance used is City-Block. Fig. 18 shows the angular error after estimating the translation vector (ϕ and β), whereas Fig. 19 shows the orientation error (θ, γ and α).

After analysing Fig. 18, it can be deduced that the translation error is smaller using a fisheye camera than a catadioptric camera with the SM. In contrast to this, the improvement with regard to the translation error can be appreciated better when the improved APOFM (all variations) is employed with the images taken by a catadioptric camera. The translation error using RANSAC (SM+RANSAC) is lower than without it (SM) and even similar to WM-SF0.6 for the fisheye images. Nevertheless, the use of the improved APOFM provides a more precise solution for all remaining cases.

With respect to the distance metric, it can be observed that the City-Block with the fisheye camera leads to a smaller translation error, independently of the proposed method case. In this sense, the behaviour with the catadioptric camera is different, the Mahalanobis distance seems to be better to the second (WM-SF0.6) and fourth (WM-LF) case, whereas the City-Block distance outputs a smaller error in the third (WM-SF0.7) and fifth (WM-SqF) case. All the same, a considerable difference of error between both distance measures can only be seen in the third case when the feature points with an associated probability higher than 0.7 are screened.

Finally, the angular error after estimating the orientation is studied. As Fig. 19 shows, the rotation error behaves very similarly to the translation error. However, it is worth highlighting that the low error

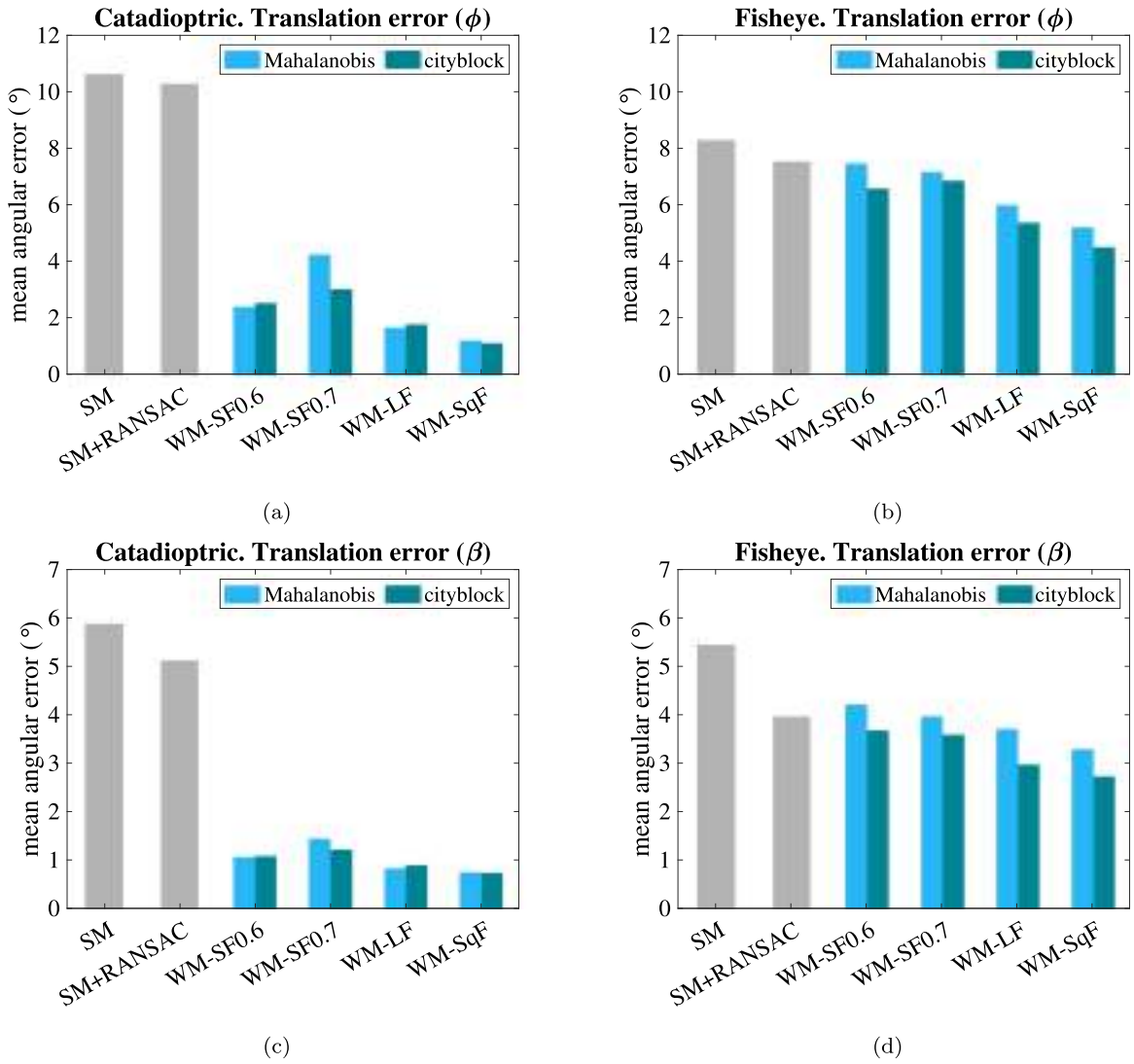


Fig. 18. Translation error. Each subfigure shows the angular error employing SM (■) and the variations of the improved APOFM (Table 1: WM-SF0.6, WM-SF0.7, WM-LF and WM-SqF) based on the distance used (Mahalanobis ■ and cityblock ■). The error estimating ϕ with a catadioptric camera is shown in (a) and with a fisheye lens in (b). The error estimating β with a catadiotric camera appears in (c) and with a fisheye lens in (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

achieved when the orientation is estimated, in contrast with the one obtained in the translation vector estimation, being this mean angular error below a remarkable value of 1°.

8. Conclusion

In this work, the localization problem is solved using visual information. The basis of this method relies on the former approach proposed in Valiente et al. (2018), which presented a visual information fusion approach for Adaptive Probability-Oriented Feature Matching (APOFM). Despite the fact that Valiente et al. (2018) exploited the potential of GP to produce a 3D representation with probability of feature existence towards obtaining a robust and adaptive matching, several aspects have been improved in the present work.

The main goal of this work was to improve the former method and to extend its application to images captured by a catadioptric and by fisheye camera, so as to produce a consistent comparison between two well-recognized vision systems within the field of visual localization. In this context, we have benchmarked the improved APOFM against: (a) a Standard Method (SM) (Hartley and Zisserman, 2003), (b) this SM with outlier rejection by means of RANSAC (SM+RANSAC) (Scaramuzza, 2011) and (c) the basic APOFM (Valiente et al., 2018) (WM-SF0.6 and WM-SF0.7). This comparative evaluation has comprised several variations associated to new contributions. This analysis appraises efficiency and computation when using these two types of vision systems under a publicly available dataset.

Additionally, we have presented an improved search method for matching candidates with the support of a k-nearest neighbour classifier which matches the nearest projected point on the images (with an associated probability) with a feature point, resulting in a matching candidate. It has been implemented as a Kd-tree algorithm using the City-block distance, and compared to an exhaustive search using the Mahalanobis distance. Next, we have improved the use of visual information in terms of the spatial probability distribution. In contrast to the previous method, which only used such information for filtering within a maximum probability those feature points allowed to result in matching candidates, now we enhance its use to achieve a weighted search of features, which dynamically adapts to the current probability of feature existence by applying three probability-weighted variations of the improved APOFM. Finally, a more reliable detection of false positives has been introduced. It is supported by a design that evaluates the pixel coincidence of the projected 3D point onto the image, assumed as a matched point between two images. Thus, a match is tagged as a false positive if the projection of its 3D point does not converge towards

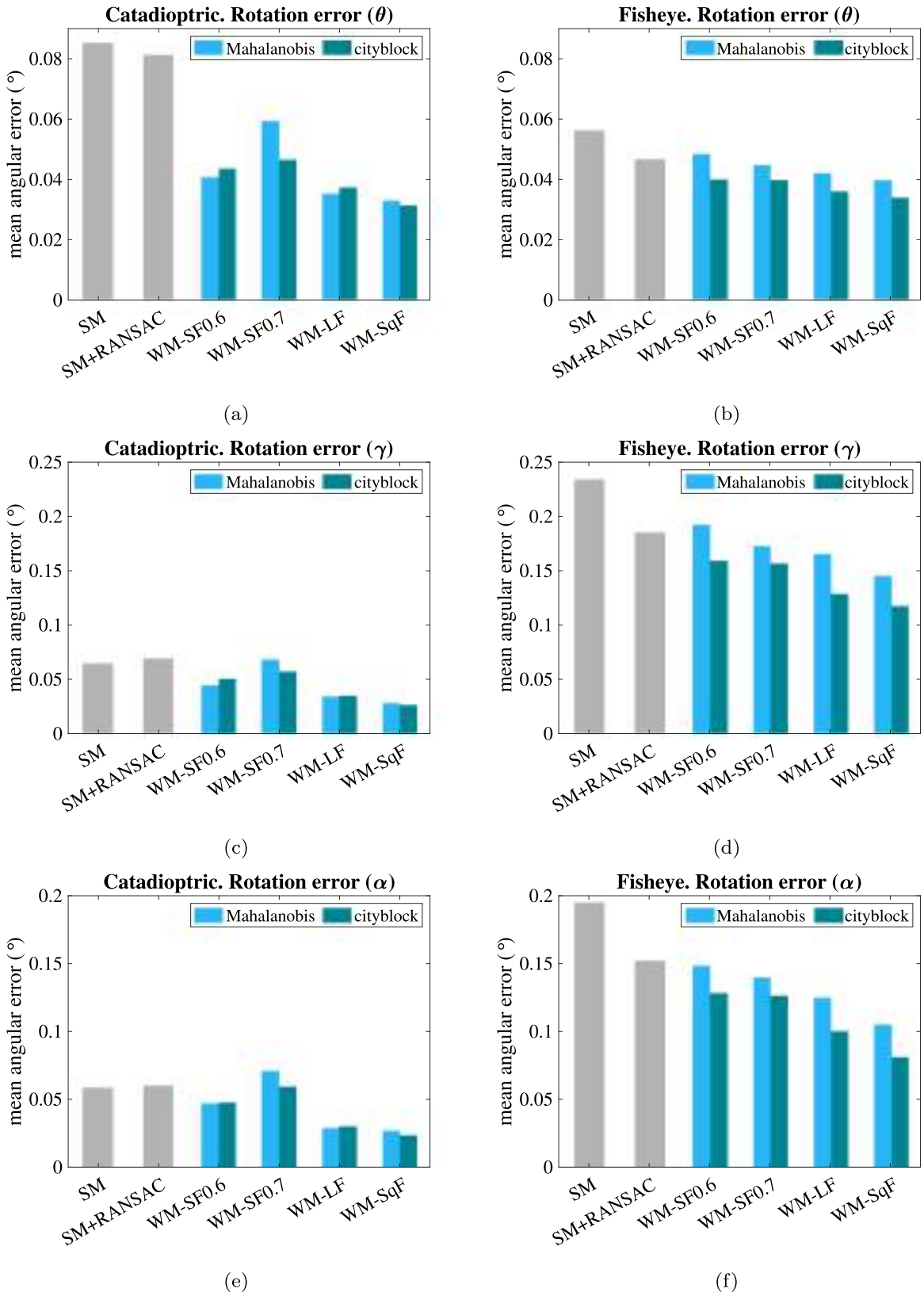


Fig. 19. Rotation error. Each subfigure shows the angular error employing SM (■) and the variations of the improved APOFM (Table 1: WM-SF0.6, WM-SF0.7, WM-LF and WM-SqF) based on the distance used (Mahalanobis ■ and cityblock ■). The error estimating θ with a catadioptric camera is shown in (a) and with a fisheye lens in (b). The error estimating γ with a catadiotric camera appears in (c) and with a fisheye lens in (d). The error estimating α with a catadiotric camera appears in (e) and with a fisheye lens in (f). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the same pixel point which was initially marked as a SURF point, within a certain distance threshold.

Before producing comparative results, a preliminary experiment was carried out in order to extract an optimum set of parameters for the

GP computation. The execution of the method constraints the accuracy with the computation load. This study reveals that a trade-off setup can be established between the spatial resolution of the 3D testing points in the probabilistic model ($\Delta grid$), and the distance threshold that discerns whether a feature point is considered as a matching candidate (χ). Although both vision systems, the catadioptric and the fisheye, should be tuned with their specific trade-off setups, these experiments considered the same value of $\Delta grid$ and χ for both vision systems test-bed, in order to ensure an acceptable balance.

After inspecting the comparative results between the catadioptric and fisheye images, it can be confirmed that the improved APOFM provides an enhanced accuracy and efficiency, comparing to SM, regardless the sort of vision system employed, either catadioptric or fisheye.

Regarding the performance of the three variations of the improved APOFM, the results corroborate several benefits of these contributions. First, they confer higher ratios of detected matching candidates, versus the total amount of feature points, in comparison with the SM. Particularly, the method WM-SF0.7 returns the highest ratio. Notably, the fisheye camera produces more matching candidates, however, due to its nonlinear nature and field of view, the final set of matched points is more reduced than the one provided by the catadioptric system.

As for the false positives detector, the proposed variations demonstrate to outperform significantly both the SM and the SM+RANSAC. The fisheye images perform better in this sense, fact that is justified by the lower number of matched points in the last stage.

Finally, focusing on the accuracy of the visual localization, it can be confirmed that the error associated to the relative pose estimation is lower when a weighted matching search with the square function is employed. Moreover, the best performance is obtained when the City-block distance is used to establish which feature point is the nearest to a certain projected point with an associated probability, and thus obtaining a valid matching candidate. The outputs of the experimental set lead to deduce that the catadioptric vision system produces lower errors with all the methods with the improved APOFM approach. It is noteworthy that the translation error, which typically is the worst affected by noise and non-linearities of these lenses, is bounded by a value under 1 degree (mean error) with the proposed variation of the method (linear and square).

In summary, this work has validated the appropriateness of the proposed contributions to deal with the visual localization problem. The estimated relative pose is defined by five angular parameters, three for the orientation and two for the translation, so this method presents the inconvenient the translation vector is obtained with the exception of the scale factor. Taking all these facts into account, the results have evidenced to outperform the SM, as well as SM+RANSAC. Furthermore, the suitability of these implementations have been extensively tested against a publicly dataset, which at the same time permitted producing an extended evaluation and comparison over two of the most commonly used vision systems in visual localization, within mobile robotics.

The evaluation has been carried out in an indoor environment. As future work, it will be interesting to employ this method using images taken from an outdoor environment. Another future work could be to compare the robustness of this method using a camera with a fisheye lens and a vision system composed of two fisheye cameras pointing to opposite sides (a full 360 degrees of view).

CRediT authorship contribution statement

María Flores: Methodology, Software, Investigation, Writing – original draft, Data curation. **David Valiente:** Methodology, Software, Supervision, Writing – review & editing, Validation. **Arturo Gil:** Methodology, Resources, Software, Writing – review & editing. **Oscar Reinoso:** Resources, Conceptualization, Validation, Project administration. **Luis Payá:** Conceptualization, Supervision, Writing – review & editing, Validation, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is part of the project PID2020-116418RB-I00 funded by MCIN/AEI/10.13039/501100011033, of the project AICO/2019/031 funded by Generalitat Valenciana, Spain, and of the grant ACIF/2020/141 funded by Generalitat Valenciana, Spain and Fondo Social Europeo, European Union.

References

Alatise, M.B., Hancke, G.P., 2020. A review on challenges of autonomous mobile robot and sensor fusion methods. IEEE Access 8, 39830–39846. http://dx.doi.org/10.1109/ACCESS.2020.2975643.

Amorós, F., Payá, L., Mayol-Cuevas, W., Jiménez, L.M., Reinoso, O., 2020. Holistic descriptors of omnidirectional color images and their performance in estimation of position and orientation. IEEE Access 8, 81822–81848. http://dx.doi.org/10.1109/ACCESS.2020.2990996.

Andert, F., Goormann, L., 2007. Combined grid and feature-based occupancy map building in large outdoor environments. In: Proceeding of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007). pp. 2065–2070. http://dx.doi.org/10.1109/IROS.2007.4399086.

Aqel, M.O.A., Marhaban, M.H., Saripan, M.I., Ismail, N.B., 2016. Review of visual odometry: types, approaches, challenges, and applications. SpringerPlus 5 (1), 1897. http://dx.doi.org/10.1186/s40064-016-3573-7.

Barone, S., Neri, P., Paoli, A., Razionale, A., 2018. Catadioptric stereo-vision system using a spherical mirror. Procedia Struct. Integr. 8, 83–91. http://dx.doi.org/10.1016/j.prostr.2017.12.010.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speeded up robust features. In: Proceedings of Computer Vision–ECCV 2006: 9th European Conference on Computer Vision. 3951, pp. 404–417. http://dx.doi.org/10.1007/11744023_32.

Beardsley, P.A., Zisserman, A., Murray, D.W., 1994. Navigation using affine structure from motion. In: Proceedings of Computer Vision – ECCV '94: Third European Conference on Computer Vision. 801, pp. 85–96. http://dx.doi.org/10.1007/BFb0028337.

Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. In: Ashenhurst, R.L. (Ed.), Commun. ACM 18 (9), 509–517. http://dx.doi.org/10.1145/361002.361007.

Boutteau, R., Savatier, X., Ertaud, J.-Y., Mazari, B., 2010. A 3D omnidirectional sensor for mobile robot applications. In: Barrera, A. (Ed.), Mobile Robots Navigation. InTech, http://dx.doi.org/10.5772/9001.

Cebollada, S., Payá, L., Mayol, W., Reinoso, O., 2019. Evaluation of clustering methods in compression of topological models and visual place recognition using global appearance descriptors. Appl. Sci. 9 (3), http://dx.doi.org/10.3390/app9030377.

Chomboon, K., Chujai, P., Teerarassammee, P., Kerdprasop, K., Kerdprasop, N., 2015. An empirical study of distance metrics for k-nearest neighbor algorithm. In: Proceedings of the 3rd International Conference on Industrial Application Engineering 2015 (ICIAE2015). pp. 280–285. http://dx.doi.org/10.12792/iciae2015.051.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27. http://dx.doi.org/10.1109/TIT.1967.1053964.

Dalla Libera, A., Tosello, E., Pillonetto, G., Ghidoni, S., Carl, R., 2019. Proprioceptive robot collision detection through Gaussian process regression. In: Proceedings of 2019 American Control Conference (ACC). pp. 19–24. http://dx.doi.org/10.23919/ACC.2019.8814361.

Emani, S., Soman, K.P., Sajith Variyar, V.V., Adarsh, S., 2019. Obstacle detection and distance estimation for autonomous electric vehicle using stereo vision and DNN. In: Wang, J., Reddy, G.R.M., Prasad, V.K., Reddy, V.S. (Eds.), Soft Computing and Signal Processing, Vol. 898. Springer Singapore, Singapore, pp. 639–648. http://dx.doi.org/10.1007/978-981-13-3393-4_65.

Fraundorfer, F., Scaramuzza, D., 2012. Visual odometry: Part II: Matching, robustness, optimization, and applications. IEEE Robot. Autom. Mag. 19 (2), 78–90. http://dx.doi.org/10.1109/MRA.2012.2182810.

Gao, W., Shen, S., 2017. Dual-fisheye omnidirectional stereo. In: Proceeding of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017). pp. 6715–6722. http://dx.doi.org/10.1109/IROS.2017.8206587.

Gao, W., Wang, K., Ding, W., Gao, F., Qin, T., Shen, S., 2020. Autonomous aerial robot using dual-fisheye cameras. J. Field Robot. 37 (4), 497–514. http://dx.doi.org/10.1002/rob.21946.

Geyer, C., Daniilidis, K., 2000. A unifying theory for central panoramic systems and practical implications. In: Proceedings of Computer Vision – ECCV 2000: 6th European Conference on Computer Vision. 1843, pp. 445–461. http://dx.doi.org/10.1007/3-540-45053-X_29.

Ghaffari, M., Gan, L., Parkison, S.A., Li, J., Eustice, R.M., 2017. Gaussian processes semantic map representation. arXiv:1707.01532 [Cs].

Ghaffari, M., Valls Miro, J., Dissanayake, G., 2018. Gaussian processes autonomous mapping and exploration for range-sensing mobile robots. Auton. Robot. 42 (2), 273–290. http://dx.doi.org/10.1007/s10514-017-9668-3.

Gil, A., Juliá, M., Reinoso, O., 2015. Occupancy grid based graph-SLAM using the distance transform, SURF features and SGD. Eng. Appl. Artif. Intell. 40, 1–10. http://dx.doi.org/10.1016/j.engappai.2014.12.010.

Harapanahalli, S., Mahony, N.O., Hernandez, G.V., Campbell, S., Riordan, D., Walsh, J., 2019. Autonomous navigation of mobile robots in factory environment. Procedia Manuf. 38, 1524–1531. http://dx.doi.org/10.1016/j.promfg.2020.01.134.

Hartley, R.I., Sturm, P., 1997. Triangulation. Comput. Vis. Image Underst. 68 (2), 146–157. http://dx.doi.org/10.1006/cviu.1997.0547.

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision, second ed. Cambridge University Press, Cambridge, UK.

Hassaballah, M., Abdelmgeid, A.A., Alshazly, H.A., 2016. Image features detection, description and matching. In: Awad, A.I., Hassaballah, M. (Eds.), Image Feature Detectors and Descriptors, Vol. 630. Springer International Publishing, Cham, pp. 11–45. http://dx.doi.org/10.1007/978-3-319-28854-3_2.

Hassaballah, M., Alshazly, H.A., Ali, A.A., 2019. Analysis and evaluation of keypoint descriptors for image matching. In: Hassaballah, M., Hosny, K.M. (Eds.), Recent Advances in Computer Vision, Vol. 804. Springer International Publishing, Cham, pp. 113–140. http://dx.doi.org/10.1007/978-3-030-03000-1_5.

Hou, J., Yu, L., Fei, S., 2020. A highly robust automatic 3D reconstruction system based on integrated optimization by point line features. Eng. Appl. Artif. Intell. 95, 103879. http://dx.doi.org/10.1016/j.engappai.2020.103879.

Jakubović, A., Velagić, J., 2018. Image feature matching and object detection using brute-force matchers. In: Proceedings of ELMAR 2018: 60th International Symposium ELMAR-2018. pp. 83–86. http://dx.doi.org/10.23919/ELMAR.2018.8534641.

Jiang, Y., Xu, Y., Liu, Y., 2013. Performance evaluation of feature detection and matching in stereo visual odometry. Neurocomputing 120, 380–390. http://dx.doi.org/10.1016/j.neucom.2012.06.055.

Jung, S., Lee, U., Jung, J., Shim, D.H., 2016. Real-time traffic sign recognition system with deep convolutional neural network. In: Proceedings of URAI 2016: 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). pp. 31–34. http://dx.doi.org/10.1109/URAI.2016.7734014.

Kostavelis, I., Charalampous, K., Gasteratos, A., Tsotsos, J.K., 2016. Robot navigation via spatial and temporal coherent semantic maps. Eng. Appl. Artif. Intell. 48, 173–187. http://dx.doi.org/10.1016/j.engappai.2015.11.004.

Lee, G.H., Fraundorfer, F., Pollefeys, M., 2013. Structureless pose-graph loop-closure with a multi-camera system on a self-driving car. In: Proceeding of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013). pp. 564–571. http://dx.doi.org/10.1109/IROS.2013.6696407.

Li, S., 2006. Full-view spherical image camera. In: Proceedings of 18th International Conference on Pattern Recognition (ICPR'06), Vol. 4. pp. 386–390. http://dx.doi.org/10.1109/ICPR.2006.585.

Li, B., Wang, Y., Zhang, Y., Zhao, W., Ruan, J., Li, P., 2020. GP-SLAM: laser-based SLAM approach based on regionalized Gaussian process map reconstruction. Auton. Robot. 44 (6), 947–967. http://dx.doi.org/10.1007/s10514-020-09906-z.

Liu, Y., Chen, J., Bai, X., 2020. An approach for multi-objective obstacle avoidance using dynamic occupancy grid map. In: Proceedings of 2020 IEEE International Conference on Mechatronics and Automation (ICMA). pp. 1209–1215. http://dx.doi.org/10.1109/ICMA49215.2020.9233760.

Liu, Y., Li, Y., Dai, L., Yang, C., Wei, L., Lai, T., Chen, R., 2021. Robust feature matching via advanced neighborhood topology consensus. Neurocomputing 421, 273–284. http://dx.doi.org/10.1016/j.neucom.2020.09.047.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60 (2), 91–110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

Marcato Junior, J., Tommaselli, A.M.G., Moraes, M.V.A., 2016. Calibration of a catadioptric omnidirectional vision system with conic mirror. ISPRS J. Photogramm. Remote Sens. 113, 97–105. http://dx.doi.org/10.1016/j.isprsjprs.2015.10.008.

Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 27 (10), 1615–1630. http://dx.doi.org/10.1109/TPAMI.2005.188.

Mohamed, S.A.S., Haghbayan, M.-H., Westerlund, T., Heikkonen, J., Tenhunen, H., Plosila, J., 2019. A survey on odometry for autonomous navigation systems. IEEE Access 7, 97466–97486. http://dx.doi.org/10.1109/ACCESS.2019.2929133.

Nair, N.S., Nair, M.S., 2020. On evolutionary computation techniques for multi-view triangulation. Mach. Vis. Appl. 31 (4), 29. http://dx.doi.org/10.1007/s00138-020-01077-2.

Nguyen, L., Miro, J.V., Shi, L., Vidal-Calleja, T., 2019. Gaussian mixture marginal distributions for modelling remaining metallic pipe wall thickness. In: Proceedings of the IEEE 2019: 9th International Conference on Cybernetics and Intelligent Systems (CIS), Robotics, Automation and Mechatronics (RAM). pp. 257–262. http://dx.doi.org/10.1109/CIS-RAM47153.2019.9095851.

Nister, D., 2003. An efficient solution to the five-point relative pose problem. In: Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03). 2, pp. II–195–202. http://dx.doi.org/10.1109/CVPR.2003.1211470.

Nutalapati, M.K., Arora, L., Bose, A., Rajawat, K., Hegde, R.M., 2019. Model free calibration of wheeled robots using Gaussian process. In: Proceedings of 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 29–35. http://dx.doi.org/10.1109/IROS40897.2019.8967569.

O'Callaghan, S.T., Ramos, F.T., 2012. Gaussian process occupancy maps. Int. J. Robot. Res. 31 (1), 42–62. http://dx.doi.org/10.1177/0278364911421039.

Park, S., Huang, Y., Goh, C.F., Shimada, K., 2018. Robot model learning with Gaussian process mixture model. In: Proceedings of 2018 IEEE: 14th International Conference on Automation Science and Engineering (CASE). pp. 1263–1268. http://dx.doi.org/10.1109/COASE.2018.8560452.

Patruno, C., Colella, R., Nitti, M., Renò, V., Mosca, N., Stella, E., 2020. A vision-based odometer for localization of omnidirectional indoor robots. Sensors 20 (3), 875. http://dx.doi.org/10.3390/s20030875.

Payá, L., Gil, A., Reinoso, O., 2017. A state-of-the-art review on mapping and localization of mobile robots using omnidirectional vision sensors. J. Sensors 2017, 1–20. http://dx.doi.org/10.1155/2017/3497650.

Poddar, S., Kottath, R., Karar, V., 2018. Evolution of visual odometry techniques. arXiv:1804.11142 [Cs].

Polymenakos, K., Laurenti, L., Patane, A., Calliess, J.-P., Cardelli, L., Kwiatkowska, M., Abate, A., Roberts, S., 2020. Safety guarantees for planning based on iterative Gaussian processes. arXiv:1912.00071 [Cs, Stat].

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. MIT Press, Cambridge, Mass.

Reinoso, O., Payá, L., 2020. Special issue on visual sensors. Sensors 20 (3), 910. http://dx.doi.org/10.3390/s20030910.

Robotics and Perception Group, University of Zurich, Switzerland, 2013. The "multi-fov" synthetic datasets. (accessed 22 October 2021), http://rpg.ifi.uzh.ch/fov.html.

Román, V., Payá, L., Cebollada, S., Reinoso, O., 2020. Creating incremental models of indoor environments through omnidirectional imaging. Appl. Sci. 10 (18), 6480. http://dx.doi.org/10.3390/app10186480.

Scaramuzza, D., 2011. 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. Int. J. Comput. Vis. 95 (1), 74–85. http://dx.doi.org/10.1007/s11263-011-0441-3.

Scaramuzza, D., 2014. Omnidirectional camera. In: Ikeuchi, K. (Ed.), Computer Vision. Springer US, Boston, MA, pp. 552–560. http://dx.doi.org/10.1007/978-0-387-31439-6_488.

Scaramuzza, D., Fraundorfer, F., 2011. Visual odometry [tutorial]. IEEE Robot. Autom. Mag. 18 (4), 80–92. http://dx.doi.org/10.1109/MRA.2011.943233.

Scaramuzza, D., Martinelli, A., Siegwart, R., 2006a. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In: Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006). p. 45. http://dx.doi.org/10.1109/ICVS.2006.3.

Scaramuzza, D., Martinelli, A., Siegwart, R., 2006b. A toolbox for easily calibrating omnidirectional cameras. In: Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5695–5701. http://dx.doi.org/10.1109/IROS.2006.282372.

Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D., 2011. Introduction to Autonomous Mobile Robots. MIT Press, Cambridge, Mass.

Song, X., Cao, Z., Gao, H., 2018. Local Gaussian processes for identifying complex mobile robot system. In: Chen, X., Zhao, Q. (Eds.), Proceedings of the 37th Chinese Control Conference. pp. 3796–3802. http://dx.doi.org/10.23919/ChiCC.2018.8483251.

Sun, K., Saulnier, K., Atanasov, N., Pappas, G.J., Kumar, V., 2018. Dense 3-D mapping with spatial correlation via Gaussian filtering. In: Proceedings of 2018 Annual American Control Conference (ACC). pp. 4267–4274. http://dx.doi.org/10.23919/ACC.2018.8431777.

Taheri, H., Xia, Z.C., 2021. SLAM; definition and evolution. Eng. Appl. Artif. Intell. 97, 104032. http://dx.doi.org/10.1016/j.engappai.2020.104032.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics. In: Intelligent robotics and autonomous agents, MIT Press, Cambridge, Mass.

Tresp, V., 2000. A Bayesian committee machine. Neural Comput. 12 (11), 2719–2741. http://dx.doi.org/10.1162/089976600300014908.

Valiente, D., Payá, L., Jiménez, L., Sebastián, J., Reinoso, O., 2018. Visual information fusion through Bayesian inference for adaptive probability-oriented feature matching. Sensors 18 (7), 2041. http://dx.doi.org/10.3390/s18072041.

Valiente García, D., Fernández Rojo, L., Gil Aparicio, A., Payá Castelló, L., Reinoso García, O., 2012. Visual odometry through appearance- and feature-based method with omnidirectional images. J. Robot. 2012, 1–13. http://dx.doi.org/10.1155/2012/797063.

Wu, B.-F., Lu, W.-C., Jen, C.-L., 2011. Monocular vision-based robot localization and target tracking. J. Robot. 2011, 1–12. http://dx.doi.org/10.1155/2011/548042.

Xiao, Q., Liu, X., Liu, M., 2012. Object tracking based on local feature matching. In: Proceedings of ISCID 2012: Fifth International Symposium on Computational Intelligence and Design (ISCID 2012). pp. 399–402. http://dx.doi.org/10.1109/ISCID.2012.106.

Yan, K., 2011. RANSAC algorithm with example of finding homography. MATLAB central file exchange. (accessed 22 October 2021), https://es.mathworks.com/matlabcentral/fileexchange/30809-ransac-algorithm-with-example-of-finding-homography.

Ying, X., Hu, Z., 2004. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In: Proceedings of Computer Vision – ECCV 2004: 8th European Conference on Computer Vision. 3021, pp. 442–455. http://dx.doi.org/10.1007/978-3-540-24670-1_34.

Yousif, K., Bab-Hadiashar, A., Hoseinnezhad, R., 2015. An overview to visual odometry and visual SLAM: Applications to mobile robotics. J. Intell. Syst. 1 (4), 289–311. http://dx.doi.org/10.1007/s40903-015-0032-7.

Yuan, M., Yau, W.-Y., Li, Z., 2018. Lost robot self-recovery via exploration using hybrid topological-metric maps. In: Proceedings of TENCON 2018: IEEE Region 10 Conference. pp. 188–193. http://dx.doi.org/10.1109/TENCON.2018.8650236.

Zhang, H., Hernandez, D., Su, Z., Su, B., 2018. A low cost vision-based road-following system for mobile robots. Appl. Sci. 8 (9), 1635. http://dx.doi.org/10.3390/app8091635.

Zhang, Z., Rebecq, H., Forster, C., Scaramuzza, D., 2016. Benefit of large field-of-view cameras for visual odometry. In: Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA 2016). pp. 801–808. http://dx.doi.org/10.1109/ICRA.2016.7487210.

Zivkovic, Z., Bakker, B., Krose, B., 2005. Hierarchical map building using visual landmarks and geometric constraints. In: Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. http://dx.doi.org/10.1109/IROS.2005.1544951, 2480–2485.

[1] D. Valiente, L. Payá, L. M. Jiménez, J. M. Sebastián y O. Reinoso, "Visual information fusion through bayesian inference for adaptive probability-oriented feature matching", *Sensors*, vol. 18, 7 2018, ISSN: 14248220. DOI: 10.3390/s18072041.

[2] M. Calonder, V. Lepetit, C. Strecha y P. Fua, "BRIEF: Binary robust independent elementary features", vol. 6314 LNCS, 2010. DOI: 10.1007/978-3-642-15561-1_56.

[3] V. Usenko, N. Demmel y D. Cremers, "The double sphere camera model", 2018. DOI: 10.1109/3DV.2018.00069.

[4] B. Khomutenko, G. Garcia y P. Martinet, "An Enhanced Unified Camera Model", *IEEE Robotics and Automation Letters*, vol. 1, 1 2016, ISSN: 23773766. DOI: 10.1109/LRA.2015.2502921.

[5] E. Rosten y T. Drummond, "Machine learning for high-speed corner detection", vol. 3951 LNCS, 2006. DOI: 10.1007/11744023_34.

[6] C. K. Williams y C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[7] J. Kannala y S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, 8 2006, ISSN: 01628828. DOI: 10.1109/TPAMI.2006.153.

[8] K. M. Yi, E. Trulls, V. Lepetit y P. Fua, "LIFT: Learned invariant feature transform", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9910 LNCS, págs. 467-483, 2016, ISSN: 16113349. DOI: 10.1007/978-3-319-46466-4_28/TABLES/3. dirección: https://link.springer.com/chapter/10.1007/978-3-319-46466-4_28.

[9] J. Engel, T. Schöps y D. Cremers, "LSD-SLAM: Large-Scale Direct monocular SLAM", vol. 8690 LNCS, 2014. DOI: 10.1007/978-3-319-10605-2_54.

[10] D. Scaramuzza, A. Martinelli y R. Siegwart, "A Toolbox for Easily Calibrating Omnidirectional Cameras", en *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, oct. de 2006, págs. 5695-5701. DOI: 10.1109/IROS.2006.282372.

[11] E. Rublee, V. Rabaud, K. Konolige y G. Bradski, "ORB: An efficient alternative to SIFT or SURF", en *2011 International Conference on Computer Vision*, 2011, págs. 2564-2571. DOI: 10.1109/ICCV.2011.6126544.

[12] Q. Zhao, W. Feng, L. Wan y J. Zhang, "SPHORB: A Fast and Robust Binary Feature on the Sphere", *International Journal of Computer Vision*, vol. 113, 2 2015, ISSN: 15731405. DOI: 10.1007/s11263-014-0787-4.

[13] Z. Wang, A. Bovik, H. Sheikh y E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Transactions on Image Processing*, vol. 13, n.º 4, págs. 600-612, abr. de 2004, ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[14] H. Bay, T. Tuytelaars y L. V. Gool, "SURF: Speeded up robust features", vol. 3951 LNCS, 2006. DOI: 10.1007/11744023_32.

[15] Y. Verdie, Kwang Moo Yi, P. Fua y V. Lepetit, "TILDE: A Temporally Invariant Learned DEtector", en *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, jun. de 2015, págs. 5279-5288, ISBN: 9781467369640. DOI: 10.1109/CVPR.2015.7299165. dirección: http://ieeexplore.ieee.org/document/7299165/.

[16] C. Mei y P. Rives, "Single view point omnidirectional camera calibration from planar grids", 2007. DOI: 10.1109/ROBOT.2007.364084.

[17] S. G. Tzafestas, "15 - Mobile Robots at Work", en *Introduction to Mobile Robot Control*, S. G. Tzafestas, ed., Oxford: Elsevier, 2014, págs. 635-663, ISBN: 978-0-12-417049-0. DOI: https://doi.org/10.1016/B978-0-12-417049-0.00015-8. dirección: https://www.sciencedirect.com/science/article/pii/B9780124170490000158.

[18] iRobot, *Robot aspirador Roomba® j7 | iRobot® | iRobot*. dirección: https://www.irobot.es/es_ES/irobot-roomba-j7/J715640.html (visitado 20-09-2023).

[19] PUDU, *BellaBot: Un robot de entrega innovador*, es. dirección: https://www.pudurobotics.com/es/products/bellabot (visitado 20-09-2023).

[20] Robotnik, *Móvil Robot RB-1 BASE - Base Robot | Robotnik®*, es-ES. dirección: https://robotnik.eu/es/productos/robots-moviles/rb-1-base/ (visitado 20-09-2023).

[21] Aldebaran, *Pepper the humanoid and programmable robot |Aldebaran*. dirección: https://www.aldebaran.com/en/pepper (visitado 20-09-2023).

[22] S. G. Tzafestas, "Mobile Robots: General Concepts", *Introduction to Mobile Robot Control*, 2014.

[23] G. Cook y F. Zhang, *Mobile robots: navigation, control and sensing, surface robots and AUVs*, eng, Second edition. Hoboken, New Jersey: Wiley, 2020, ISBN: 9781119534785.

[24] R. Kala, "Chapter2: Localization and mapping", en Elsevier Science, 2023, págs. 635-663, ISBN: 978-0-443-189098-1.

[25] O. Liu, S. Yuan y Z. Li, "A Survey on Sensor Technologies for Unmanned Ground Vehicles", en *2020 3rd International Conference on Unmanned Systems (ICUS)*, 2020, págs. 638-645. DOI: 10.1109/ICUS50048.2020.9274845.

[26] M. A. de Miguel, F. García y J. M. Armingol, "Improved LiDAR Probabilistic Localization for Autonomous Vehicles Using GNSS", *Sensors*, vol. 20, n.º 11, 2020, ISSN: 1424-8220. DOI: 10.3390/s20113145. dirección: https://www.mdpi.com/1424-8220/20/11/3145.

[27] L. Li, R. Wang y X. Zhang, *A Tutorial Review on Point Cloud Registrations: Principle, Classification, Comparison, and Technology Challenges*, 2021. DOI: 10.1155/2021/9953910.

[28] F. Fraundorfer y D. Scaramuzza, "Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications", *IEEE Robotics Automation Magazine*, vol. 19, n.º 2, págs. 78-90, 2012.

[29] D. Scaramuzza y F. Fraundorfer, "Visual Odometry [Tutorial]", *IEEE Robotics Automation Magazine*, vol. 18, n.º 4, págs. 80-92, 2011.

[30] M. O. Aqel, M. H. Marhaban, M. I. Saripan y N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications", *SpringerPlus*, vol. 5, n.º 1, pág. 1897, 2016.

[31] S. Suzuki y R. Suda, "A vision system with wide field of view and collision alarms for teleoperation of mobile robots", *ROBOMECH Journal*, vol. 1, 1 2014, ISSN: 21974225. DOI: 10.1186/s40648-014-0008-5.

[32] Y. Yang, D. Tang, D. Wang, W. Song, J. Wang y M. Fu, "Multi-camera visual SLAM for off-road navigation", *Robotics and Autonomous Systems*, vol. 128, pág. 103 505, 2020, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2020.103505. dirección: https://www.sciencedirect.com/science/article/pii/S0921889019308711.

[33] D. Valiente, A. Gil, L. Payá, J. M. Sebastián y O. Reinoso, "Robust visual localization with dynamic uncertainty management in omnidirectional SLAM", *Applied Sciences (Switzerland)*, vol. 7, 12 2017, ISSN: 20763417. DOI: 10.3390/app7121294.

[34] S. Liu, P. Guo, L. Feng y A. Yang, "Accurate and robust monocular SLAM with omnidirectional cameras", *Sensors (Switzerland)*, vol. 19, 20 2019, ISSN: 14248220. DOI: 10.3390/s19204494.

[35] R. Mur-Artal, J. M. Montiel y J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System", *IEEE Transactions on Robotics*, vol. 31, págs. 1147-1163, 5 oct. de 2015, ISSN: 15523098. DOI: 10.1109/TRO.2015.2463671.

[36] R. Mur-Artal y J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras", *IEEE Transactions on Robotics*, vol. 33, págs. 1255-1262, 5 oct. de 2017, ISSN: 15523098. DOI: 10.1109/TRO.2017.2705103.

[37] L. Yao, Y. Lin, C. Zhu y Z. Wang, "An Effective Dual-Fisheye Lens Stitching Method Based on Feature Points", vol. 11295 LNCS, 2019. DOI: 10.1007/978-3-030-05710-7_55.

[38] V. Usenko, N. Demmel, D. Schubert, J. Stuckler y D. Cremers, "Visual-Inertial Mapping with Non-Linear Factor Recovery", *IEEE Robotics and Automation Letters*, vol. 5, 2 2020, ISSN: 23773766. DOI: 10.1109/LRA.2019.2961227.

[39] Zichao Zhang, H. Rebecq, C. Forster y D. Scaramuzza, "Benefit of large field-of-view cameras for visual odometry", en *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden: IEEE, mayo de 2016, págs. 801-808, ISBN: 9781467380263. DOI: 10.1109/ICRA.2016.7487210.

[40] ARVC, *Laboratorio de Automatización Robótica y Visión por Computador (ARVC) - UMH*, https://arvc.umh.es/db/360views/, Online; accessed 16 February 2023.

[41] P. F. Alcantarilla, A. Bartoli y A. J. Davison, "KAZE Features", en *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato y C. Schmid, eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, págs. 214-227, ISBN: 978-3-642-33783-3.

[42] R. Szeliski, "Image Formation", en *Computer Vision: Algorithms and Applications*. Cham: Springer International Publishing, 2022, págs. 27-83, ISBN: 978-

3-030-34372-9. DOI: 10.1007/978-3-030-34372-9_2. dirección: https://doi.org/10.1007/978-3-030-34372-9_2.

[43] A. S. Aguiar, F. N. dos Santos, J. B. Cunha, H. Sobreira y A. J. Sousa, "Localization and Mapping for Robots in Agriculture and Forestry: A Survey", *Robotics*, vol. 9, n.º 4, 2020, ISSN: 2218-6581. DOI: 10.3390/robotics9040097. dirección: https://www.mdpi.com/2218-6581/9/4/97.

[44] S. Ekvall, D. Kragic y P. Jensfelt, "Object detection and mapping for service robot tasks", vol. 25, 2007. DOI: 10.1017/S0263574706003237.

[45] D. Chatziparaschis, M. G. Lagoudakis y P. Partsinevelos, "Aerial and ground robot collaboration for autonomous mapping in search and rescue missions", *Drones*, vol. 4, págs. 1-24, 4 dic. de 2020, ISSN: 2504446X. DOI: 10.3390/DRONES4040079.

[46] Y. Zhang, J. Huang y L. Han, *Research status of planetary surface mobile exploration robots: Review*, ene. de 2021. DOI: 10.7527/S1000-6893.2020.23909.

[47] C. Papachristos, S. Khattak, F. Mascarich y K. Alexis, "Autonomous Navigation and Mapping in Underground Mines Using Aerial Robots", en *2019 IEEE Aerospace Conference*, 2019, págs. 1-8. DOI: 10.1109/AERO.2019.8741532.

[48] D. Lee, G. Kang, B. Kim y D. H. Shim, "Assistive Delivery Robot Application for Real-World Postal Services", *IEEE Access*, vol. 9, págs. 141 981-141 998, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3120618.

[49] J. Lee et al., "ODS-Bot: Mobile Robot Navigation for Outdoor Delivery Services", *IEEE Access*, vol. 10, págs. 107 250-107 258, 2022, ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3212768.

[50] T. Tanioka, *Nursing and rehabilitative care of the elderly using humanoid robots*, 2019. DOI: 10.2152/jmi.66.19.

[51] F. Tanaka, K. Isshiki, F. Takahashi, M. Uekusa, R. Sei y K. Hayashi, "Pepper learns together with children: Development of an educational application", vol. 2015-December, 2015. DOI: 10.1109/HUMANOIDS.2015.7363546.

[52] K. Blöcher y R. Alt, "AI and robotics in the European restaurant sector: Assessing potentials for process innovation in a high-contact service industry", *Electronic Markets*, vol. 31, págs. 529-551, 3 sep. de 2021, ISSN: 14228890. DOI: 10.1007/s12525-020-00443-2.

[53] J. M. Garcia-Haro, E. D. Oña, J. Hernandez-Vicen, S. Martinez y C. Balaguer, *Service robots in catering applications: A review and future challenges*, 2021. DOI: 10.3390/electronics10010047.

[54] J. Zhang, Y. Ou, G. Jiang e Y. Zhou, "An approach to restaurant service robot SLAM", 2016. DOI: 10.1109/ROBIO.2016.7866643.

[55] H. T. Tran et al., "A novel design of a smart interactive guiding robot for busy airports", *International Journal on Smart Sensing and Intelligent Systems*, vol. 15, 1 ene. de 2022, ISSN: 11785608. DOI: 10.2478/ijssis-2022-0017.

[56] R. Triebel et al., "SPENCER: A socially aware service robot for passenger guidance and help in busy airports", vol. 113, 2016. DOI: 10.1007/978-3-319-27702-8_40.

[57] A. K. Bordoloi, F. Islam, J. Zaman, N. Phukan y N. M. Kakoty, "A floor cleaning robot for domestic environments", vol. Part F132085, Association for

Computing Machinery, jun. de 2017, ISBN: 9781450352949. DOI: 10.1145/3132446.3134883.

[58] K. Yovchev, D. Chikurtev, N. Chivarov y M. Grueva, "An intelligent control system for service robots", vol. 52, 2019. DOI: 10.1016/j.ifacol.2019.12.544.

[59] A. Eirale, M. Martini, L. Tagliavini, D. Gandini, M. Chiaberge y G. Quaglia, "Marvin: An Innovative Omni-Directional Robotic Assistant for Domestic Environments", *Sensors*, vol. 22, 14 2022, ISSN: 14248220. DOI: 10.3390/s22145261.

[60] F. Rubio, F. Valero y C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications", DOI: 10.1177/1729881419839596. dirección: https://us.sagepub.com/en-us/nam/.

[61] M. A. Niloy et al., "Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review", *IEEE Access*, vol. 9, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3062557.

[62] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age", *IEEE Transactions on Robotics*, vol. 32, 6 2016, ISSN: 15523098. DOI: 10.1109/TRO.2016.2624754.

[63] X. Zhou y R. Huang, "A State-of-the-Art Review on SLAM", en *Intelligent Robotics and Applications*, H. Liu et al., eds., Cham: Springer International Publishing, 2022, págs. 240-251, ISBN: 978-3-031-13835-5.

[64] N. Adzhar, Y. Yusof y M. A. Ahmad, "A review on autonomous mobile robot path planning algorithms", *Advances in Science, Technology and Engineering Systems*, vol. 5, 3 2020, ISSN: 24156698. DOI: 10.25046/aj050330.

[65] J. R. Sánchez-Ibáñez, C. J. Pérez-Del-pulgar y A. García-Cerezo, *Path planning for autonomous mobile robots: A review*, 2021. DOI: 10.3390/s21237898.

[66] I. S. Ramírez, P. J. Bernalte Sánchez, M. Papaelias y F. P. G. Márquez, "Autonomous Underwater Vehicles and Field of View in Underwater Operations", *Journal of Marine Science and Engineering*, vol. 9, n.º 3, 2021, ISSN: 2077-1312. DOI: 10.3390/jmse9030277. dirección: https://www.mdpi.com/2077-1312/9/3/277.

[67] W. Chen et al., *SLAM Overview: From Single Sensor to Heterogeneous Fusion*, 2022. DOI: 10.3390/rs14236033.

[68] G. Jia et al., *Visual-SLAM Classical Framework and Key Techniques: A Review*, 2022. DOI: 10.3390/s22124582.

[69] A. Tourani, H. Bavle, J. L. Sanchez-Lopez y H. Voos, *Visual SLAM: What Are the Current Trends and What to Expect?*, 2022. DOI: 10.3390/s22239297.

[70] Y. Zhang, Y. Wu, K. Tong, H. Chen e Y. Yuan, "Review of Visual Simultaneous Localization and Mapping Based on Deep Learning", *Remote Sensing*, vol. 15, n.º 11, pág. 2740, mayo de 2023, ISSN: 2072-4292. DOI: 10.3390/rs15112740. dirección: https://www.mdpi.com/2072-4292/15/11/2740.

[71] J. Crespo, J. C. Castillo, O. M. Mozos y R. Barber, "Semantic Information for Robot Navigation: A Survey", *Applied Sciences 2020, Vol. 10, Page 497*, vol. 10, pág. 497, 2 ene. de 2020, ISSN: 2076-3417. DOI: 10.3390/APP10020497. dirección: https://www.mdpi.com/2076-3417/10/2/497/htmhttps://www.mdpi.com/2076-3417/10/2/497.

[72] W. Chen et al., *An Overview on Visual SLAM: From Tradition to Semantic*, 2022. DOI: 10.3390/rs14133010.

[73] H. Andreasson, G. Grisetti, T. Stoyanov, A. Pretto y A. Ruberti, "Sensors for Mobile Robots", *Encyclopedia of Robotics*, págs. 1-22, 2023. DOI: 10.1007/978-3-642-41610-1_159-1.

[74] D. F. Pierrottet, F. Amzajerdian, G. D. Hines, B. W. Barnes, L. B. Petway y J. M. Carson, "Lidar development at NASA langley research center for vehicle navigation and landing in GPS denied environments", 2018. DOI: 10.1109/RAPID.2018.8508958.

[75] Q. Yang, D. Qu, F. Xu, F. Zou, G. He y M. Sun, "Mobile robot motion control and autonomous navigation in GPS-denied outdoor environments using 3D laser scanning", *Assembly Automation*, vol. 39, págs. 469-478, 3 ago. de 2019, ISSN: 01445154. DOI: 10.1108/AA-02-2018-029.

[76] K. Yamada, S. Koga, T. Shimoda y K. Sato, "Autonomous Path Travel Control of Mobile Robot Using Internal and External Camera Images in GPS-Denied Environments", *Journal of Robotics and Mechatronics*, vol. 33, págs. 1284-1293, 6 dic. de 2021, ISSN: 18838049. DOI: 10.20965/jrm.2021.p1284.

[77] M. Aftatah, A. Lahrech, A. Abounada y A. Soulhi, "GPS/INS/Odometer Data Fusion for Land Vehicle Localization in GPS Denied Environment", *Modern Applied Science*, vol. 11, pág. 62, 1 oct. de 2016, ISSN: 1913-1844. DOI: 10.5539/mas.v11n1p62.

[78] J. Patoliya, H. Mewada, M. Hassaballah, M. A. Khan y S. Kadry, "A robust autonomous navigation and mapping system based on GPS and LiDAR data for unconstraint environment", *Earth Science Informatics*, vol. 15, págs. 2703-2715, 4 dic. de 2022, ISSN: 18650481. DOI: 10.1007/S12145-022-00791-X/TABLES/2. dirección: https://link.springer.com/article/10.1007/s12145-022-00791-x.

[79] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard y D. Nardi, "Fast and accurate SLAM with Rao–Blackwellized particle filters", *Robotics and Autonomous Systems*, vol. 55, n.º 1, págs. 30-38, 2007.

[80] C. Jung y W. Chung, "Calibration of kinematic parameters for two wheel differential mobile robots by using experimental heading errors", *International Journal of Advanced Robotic Systems*, vol. 8, págs. 134-142, 6 2011, ISSN: 17298814. DOI: 10.5772/50906.

[81] U. Onyekpe, V. Palade, A. Herath, S. Kanarachos y M. E. Fitzpatrick, "WhONet: Wheel Odometry neural Network for vehicular localisation in GNSS-deprived environments", *Engineering Applications of Artificial Intelligence*, vol. 105, 2021, ISSN: 09521976. DOI: 10.1016/j.engappai.2021.104421.

[82] G. G. Samatas y T. P. Pachidis, *Inertial Measurement Units (IMUs) in Mobile Robots over the Last Five Years: A Review*, feb. de 2022. DOI: 10.3390/designs6010017.

[83] K. Yan y B. Ma, "Obstacle Avoidance Based on 2D-Lidar in Unknown Environment", vol. 592, Springer Verlag, 2020, págs. 609-618, ISBN: 9789813296817. DOI: 10.1007/978-981-32-9682-4_64.

[84] A. Asvadi, C. Premebida, P. Peixoto y U. Nunes, "3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels

and multi-region ground planes", *Robotics and Autonomous Systems*, vol. 83, 2016, ISSN: 09218890. DOI: 10.1016/j.robot.2016.06.007.

[85] M. Sarkar, M. Prabhakar y D. Ghose, "Avoiding Obstacles with Geometric Constraints on LiDAR Data for Autonomous Robots", vol. 608, 2023. DOI: 10.1007/978-981-19-9225-4_54.

[86] A. Cherubini, F. Spindler y F. Chaumette, "Autonomous visual navigation and laser-based moving obstacle avoidance", *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, págs. 2101-2110, 5 oct. de 2014, ISSN: 15249050. DOI: 10.1109/TITS.2014.2308977.

[87] S. Y. Alaba y J. E. Ball, *A Survey on Deep-Learning-Based LiDAR 3D Object Detection for Autonomous Driving*, 2022. DOI: 10.3390/s22249577.

[88] S. Shi, X. Wang y H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud", vol. 2019-June, 2019. DOI: 10.1109/CVPR.2019.00086.

[89] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby y A. Mouzakitis, "A Survey on 3D Object Detection Methods for Autonomous Driving Applications", *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, 10 2019, ISSN: 15580016. DOI: 10.1109/TITS.2019.2892405.

[90] J. Zhang y S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time", MIT Press Journals, 2014. DOI: 10.15607/RSS.2014.X.007.

[91] Q. Zou, Q. Sun, L. Chen, B. Nie y Q. Li, "A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, 7 2022, ISSN: 15580016. DOI: 10.1109/TITS.2021.3063477.

[92] S. Liang, Z. Cao, C. Wang y J. Yu, "A Novel 3D LiDAR SLAM Based on Directed Geometry Point and Sparse Frame", *IEEE Robotics and Automation Letters*, vol. 6, 2 2021, ISSN: 23773766. DOI: 10.1109/LRA.2020.3043200.

[93] A. Nüchter, K. Lingemann, J. Hertzberg y H. Surmann, "6D SLAM - 3D mapping outdoor environments", vol. 24, 2007. DOI: 10.1002/rob.20209.

[94] Y. Pan, P. Xiao, Y. He, Z. Shao y Z. Li, "Mulls: Versatile LiDAR SLAM via Multi-metric Linear Least Square", vol. 2021-May, 2021. DOI: 10.1109/ICRA48506.2021.9561364.

[95] D. Cattaneo, M. Vaghi y A. Valada, "LCDNet: Deep Loop Closure Detection and Point Cloud Registration for LiDAR SLAM", *IEEE Transactions on Robotics*, vol. 38, 4 2022, ISSN: 19410468. DOI: 10.1109/TRO.2022.3150683.

[96] T. Liu, Y. Wang, X. Niu, L. Chang, T. Zhang y J. Liu, "LiDAR Odometry by Deep Learning-Based Feature Points with Two-Step Pose Estimation", *Remote Sensing*, vol. 14, 12 jun. de 2022, ISSN: 20724292. DOI: 10.3390/rs14122764.

[97] M. Jokela, M. Kutila y P. Pyykönen, "Testing and validation of automotive point-cloud sensors in adverse weather conditions", *Applied Sciences (Switzerland)*, vol. 9, 11 2019, ISSN: 20763417. DOI: 10.3390/app9112341.

[98] Z. Hong, Y. Petillot y S. Wang, "RadarSLAM: Radar based large-scale SLAM in all weathers", 2020. DOI: 10.1109/IROS45743.2020.9341287.

[99] C. Q. Zhao, Q. Y. Sun, C. Z. Zhang, Y. Tang y F. Qian, *Monocular depth estimation based on deep learning: An overview*, 2020. DOI: 10.1007/s11431-020-1582-8.

[100] F. Khan, S. Salahuddin y H. Javidnia, *Deep learning-based monocular depth estimation methods—a state-of-the-art review*, 2020. DOI: 10.3390/s20082272.

[101] C. Debeunne y D. Vivet, "A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping", *Sensors 2020, Vol. 20, Page 2068*, vol. 20, pág. 2068, 7 abr. de 2020, ISSN: 1424-8220. DOI: 10.3390/S20072068. dirección: https://www.mdpi.com/1424-8220/20/7/2068/htmhttps://www.mdpi.com/1424-8220/20/7/2068.

[102] W. Chen, W. Tian, X. Xie y W. Stork, "RGB Image- and Lidar-Based 3D Object Detection Under Multiple Lighting Scenarios", *Automotive Innovation*, vol. 5, 3 2022, ISSN: 25228765. DOI: 10.1007/s42154-022-00176-2.

[103] V. D. Silva, J. Roche y A. Kondoz, "Robust Fusion of LiDAR and Wide-Angle Camera Data for Autonomous Mobile Robots", *Sensors 2018, Vol. 18, Page 2730*, vol. 18, pág. 2730, 8 ago. de 2018, ISSN: 1424-8220. DOI: 10.3390/S18082730. dirección: https://www.mdpi.com/1424-8220/18/8/2730/htmhttps://www.mdpi.com/1424-8220/18/8/2730.

[104] H. Y. Lin y X. Z. Peng, "Autonomous Quadrotor Navigation with Vision Based Obstacle Avoidance and Path Planning", *IEEE Access*, vol. 9, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3097945.

[105] D. Ball et al., "Vision-based Obstacle Detection and Navigation for an Agricultural Robot", *Journal of Field Robotics*, vol. 33, 8 2016, ISSN: 15564967. DOI: 10.1002/rob.21644.

[106] N. Rezaei y S. Darabi, "Mobile robot monocular vision-based obstacle avoidance algorithm using a deep neural network", *Evolutionary Intelligence*, 2023, ISSN: 18645917. DOI: 10.1007/s12065-023-00829-z.

[107] Y. Wei, W. Wei e Y. Zhang, "EfferDeepNet: An Efficient Semantic Segmentation Method for Outdoor Terrain", *Machines*, vol. 11, 2 2023, ISSN: 20751702. DOI: 10.3390/machines11020256.

[108] S. Chen, D. Shao, L. Zhang y C. Zhang, "Learning depth-aware features for indoor scene understanding", *Multimedia Tools and Applications*, vol. 81, 29 2022, ISSN: 15737721. DOI: 10.1007/s11042-021-11453-3.

[109] Y. Wang, Q. Chen, S. Chen y J. Wu, "Multi-Scale Convolutional Features Network for Semantic Segmentation in Indoor Scenes", *IEEE Access*, vol. 8, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2993570.

[110] L. Li, Z. Liu, Ümit Özgüner, J. Lian, Y. Zhou e Y. Zhao, "Dense 3D Semantic SLAM of traffic environment based on stereo vision", vol. 2018-June, 2018. DOI: 10.1109/IVS.2018.8500714.

[111] H. Tang, H. Zhu, L. Fei, T. Wang, Y. Cao y C. Xie, "Low-Illumination Image Enhancement Based on Deep Learning Techniques: A Brief Review", *Photonics*, vol. 10, 2 feb. de 2023, ISSN: 23046732. DOI: 10.3390/photonics10020198.

[112] M. Aladem, S. Baek y S. A. Rawashdeh, "Evaluation of Image Enhancement Techniques for Vision-Based Navigation under Low Illumination", *Journal of Robotics*, vol. 2019, 2019, ISSN: 16879619. DOI: 10.1155/2019/5015741.

[113] S. Cebollada, L. Payá, V. Román y O. Reinoso, "Hierarchical Localization in Topological Models Under Varying Illumination Using Holistic Visual Descriptors", *IEEE Access*, vol. 7, págs. 49 580-49 595, 2019. DOI: 10.1109/ACCESS.2019.2910581.

[114] K. M. Othman y A. B. Rad, "A Doorway Detection and Direction (3Ds) System for Social Robots via a Monocular Camera", *Sensors 2020, Vol. 20, Page 2477*, vol. 20, pág. 2477, 9 abr. de 2020, ISSN: 1424-8220. DOI: 10.3390/S20092477. dirección: https://www.mdpi.com/1424-8220/20/9/2477/htmhttps://www.mdpi.com/1424-8220/20/9/2477.

[115] Y. Hirakawa e Y. Kuroda, "Monocular Navigation System for Corridor Environments Based on Relative Camera Pose Estimation: An Approach Without SLAM", Institute of Electrical y Electronics Engineers Inc., 2023, ISBN: 9798350398687. DOI: 10.1109/SII55687.2023.10039123.

[116] A. J. Davison, I. D. Reid, N. D. Molton y O. Stasse, "MonoSLAM: Real-time single camera SLAM", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, 6 2007, ISSN: 01628828. DOI: 10.1109/TPAMI.2007.1049.

[117] D. C. Herath, S. Kodagoda y G. Dissanayake, "Simultaneous localisation and mapping: A stereo vision based approach", 2006. DOI: 10.1109/IROS.2006.281749.

[118] F. Wang, E. Lü, Y. Wang, G. Qiu y H. Lu, "Efficient stereo visual simultaneous localization and mapping for an autonomous unmanned forklift in an unstructured warehouse", *Applied Sciences (Switzerland)*, vol. 10, 2 2020, ISSN: 20763417. DOI: 10.3390/app10020698.

[119] D. Esparza y G. Flores, "The STDyn-SLAM: A Stereo Vision and Semantic Segmentation Approach for VSLAM in Dynamic Outdoor Environments", *IEEE Access*, vol. 10, 2022, ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3149885.

[120] J. Mo, M. J. Islam y J. Sattar, "Fast Direct Stereo Visual SLAM", *IEEE Robotics and Automation Letters*, vol. 7, págs. 778-785, 2 abr. de 2022, ISSN: 23773766. DOI: 10.1109/LRA.2021.3133860.

[121] S. Badrloo, M. Varshosaz, S. Pirasteh y J. Li, *Image-Based Obstacle Detection Methods for the Safe Navigation of Unmanned Vehicles: A Review*, ago. de 2022. DOI: 10.3390/rs14153824.

[122] C. Wang, T. Wang, J. Liang, Y. Chen, Y. Zhang y C. Wang, "Monocular visual SLAM for small UAVs in GPS-denied environments", en *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012, págs. 896-901. DOI: 10.1109/ROBIO.2012.6491082.

[123] A. Al-Kaff, F. García, D. Martín, A. de la Escalera y J. M. Armingol, "Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs", *Sensors (Switzerland)*, vol. 17, 5 mayo de 2017, ISSN: 14248220. DOI: 10.3390/s17051061.

[124] M. He, C. Zhu, Q. Huang, B. Ren y J. Liu, "A review of monocular visual odometry", *Visual Computer*, vol. 36, págs. 1053-1065, 5 mayo de 2020, ISSN: 01782789. DOI: 10.1007/S00371-019-01714-6/FIGURES/6. dirección: https://link.springer.com/article/10.1007/s00371-019-01714-6.

[125] D. Scaramuzza, "Omnidirectional Camera", en *Computer Vision: A Reference Guide*, K. Ikeuchi, ed. Boston, MA: Springer US, 2014, págs. 552-560, ISBN: 978-0-387-31439-6. DOI: 10.1007/978-0-387-31439-6_488. dirección: https://doi.org/10.1007/978-0-387-31439-6_488.

[126] H. Araujo, "Omnidirectional Vision", *Encyclopedia of Robotics*, págs. 1-9, 2020. DOI: 10.1007/978-3-642-41610-1_101-1. dirección: https://link.springer.com/referenceworkentry/10.1007/978-3-642-41610-1_101-1.

[127] K. Chappellet, G. Caron, F. Kanehiro, K. Sakurada y A. Kheddar, "Benchmarking cameras for OpenVSLAM indoors", *Proceedings - International Conference on Pattern Recognition*, págs. 4857-4864, 2020, ISSN: 10514651. DOI: 10.1109/ICPR48806.2021.9413278.

[128] I. Tošić y P. Frossard, "CHAPTER 10 - Spherical Imaging in Omnidirectional Camera Networks", en *Multi-Camera Networks*, H. Aghajan y A. Cavallaro, eds., Oxford: Academic Press, 2009, págs. 239-264, ISBN: 978-0-12-374633-7. DOI: https://doi.org/10.1016/B978-0-12-374633-7.00012-4. dirección: https://www.sciencedirect.com/science/article/pii/B9780123746337000124.

[129] S. G. Tzafestas, "4 - Mobile Robot Sensors", en *Introduction to Mobile Robot Control*, S. G. Tzafestas, ed., Oxford: Elsevier, 2014, págs. 101-135, ISBN: 978-0-12-417049-0. DOI: https://doi.org/10.1016/B978-0-12-417049-0.00004-3. dirección: https://www.sciencedirect.com/science/article/pii/B9780124170490000043.

[130] S. Gao et al., "Compact and lightweight panoramic annular lens for computer vision tasks", *Optics Express*, vol. 30, 17 2022, ISSN: 10944087. DOI: 10.1364/oe.465888.

[131] J. Bai et al., "PALVO: visual odometry based on panoramic annular lens", *Optics Express, Vol. 27, Issue 17, pp. 24481-24497*, vol. 27, págs. 24 481-24 497, 17 ago. de 2019, ISSN: 1094-4087. DOI: 10.1364/OE.27.024481.

[132] D. Wang, J. Wang, Y. Tian, K. Hu y M. Xu, "PAL-SLAM: a feature-based SLAM system for a panoramic annular lens", *Optics Express*, vol. 30, 2 2022, ISSN: 10944087. DOI: 10.1364/oe.447893.

[133] Y. Fang, K. Yang, R. Cheng, L. Sun y K. Wang, "A panoramic localizer based on coarse-to-fine descriptors for navigation assistance", *Sensors (Switzerland)*, vol. 20, 15 2020, ISSN: 14248220. DOI: 10.3390/s20154177.

[134] Insta360, *Insta360 Pro - Transporta a tu audiencia: RV en 8K*, es-es. dirección: https://www.insta360.com/es/product/insta360-pro/#pro_top (visitado 28-09-2023).

[135] FLIR, *Ladybug6 | Teledyne FLIR*, es-ES. dirección: https://www.flir.es/products/ladybug6?vertical=machine+vision&segment=iis (visitado 28-09-2023).

[136] R. THETA, *Ricoh Theta Camera 360° | RICOH THETA Z1*. dirección: https://ricohtheta.eu/products/ricoh-theta-z1 (visitado 28-09-2023).

[137] Garmin, *VIRB 360*, es-ES. dirección: https://www.garmin.com/es-ES/p/562010 (visitado 28-09-2023).

[138] J. Dong, H. Yu y L. Kong, "Research on imaging model and unwrapping algorithm of catadioptrio-mnidirectional vision system", 2018. DOI: 10.1109/ICMA.2018.8484410.

[139] N. S. Chong, Y. H. Kho y M. L. D. Wong, "A closed form unwrapping method for a spherical omnidirectional view sensor", *EURASIP Journal on Image and*

Video Processing, vol. 2013, 2013, ISSN: 16875176. DOI: 10.1186/1687-5281-2013-5.

[140] R. Szeliski, "Image Alignment and Stitching", en *Computer Vision: Algorithms and Applications*. Cham: Springer International Publishing, 2022, págs. 401-441, ISBN: 978-3-030-34372-9. DOI: 10.1007/978-3-030-34372-9_8. dirección: https://doi.org/10.1007/978-3-030-34372-9_8.

[141] R. Szeliski, *Image Alignment and Stitching: A Tutorial* (Foundations and trends in computer graphics and vision). now publishers, 2006, vol. 2, ISBN: 9781933019048. DOI: 10.1561/0600000009.

[142] G. Meneghetti, M. Danelljan, M. Felsberg y K. Nordberg, "Image alignment for panorama stitching in sparsely structured environments", vol. 9127, 2015. DOI: 10.1007/978-3-319-19665-7_36.

[143] M. Jagadeeswari, C. S. Manikandababu, M. S. Dhviya y J. V. Meenakshi, "Review: A Comparative Study based on Video Stitching Methods", *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021*, págs. 1157-1163, ago. de 2021. DOI: 10.1109/ICESC51422.2021.9532635.

[144] A. Pandey y U. C. Pati, "Image mosaicing: A deeper insight", *Image and Vision Computing*, vol. 89, 2019, ISSN: 02628856. DOI: 10.1016/j.imavis.2019.07.002.

[145] E. Adel, M. Elmogy y H. Elbakry, "Image Stitching based on Feature Extraction Techniques: A Survey", *International Journal of Computer Applications*, vol. 99, 6 2014. DOI: 10.5120/17374-7818.

[146] S. K. Sharma, K. Jain y A. K. Shukla, "A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching", *Applied Sciences*, vol. 13, n.º 10, 2023, ISSN: 2076-3417. DOI: 10.3390/app13106015. dirección: https://www.mdpi.com/2076-3417/13/10/6015.

[147] D. DeTone, T. Malisiewicz y A. Rabinovich, *Deep Image Homography Estimation*, arXiv:1606.03798 [cs], jun. de 2016. DOI: 10.48550/arXiv.1606.03798. dirección: http://arxiv.org/abs/1606.03798 (visitado 06-06-2023).

[148] N. Yan et al., *Deep Learning on Image Stitching With Multi-viewpoint Images: A Survey*, 2023. DOI: 10.1007/s11063-023-11226-z.

[149] M. Alomran y D. Chai, "Feature-based panoramic image stitching", en *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016, págs. 1-6. DOI: 10.1109/ICARCV.2016.7838721.

[150] T. Ho y M. Budagavi, "Dual-fisheye lens stitching for 360-degree imaging", en *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA: IEEE, mar. de 2017, págs. 2172-2176, ISBN: 9781509041176. DOI: 10.1109/ICASSP.2017.7952541.

[151] T. Ho, I. D. Schizas, K. R. Rao y M. Budagavi, "360-degree video stitching for dual-fisheye lens cameras based on rigid moving least squares", en *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing: IEEE, sep. de 2017, págs. 51-55, ISBN: 9781509021758. DOI: 10.1109/ICIP.2017.8296241.

[152] T. Souza et al., "360 Stitching from Dual-Fisheye Cameras Based on Feature Cluster Matching", en *2018 31st SIBGRAPI Conference on Graphics, Patterns*

and Images (SIBGRAPI), Parana: IEEE, oct. de 2018, págs. 313-320, ISBN: 9781538692646. DOI: 10.1109/SIBGRAPI.2018.00047.

[153] G. Ni, X. Chen, Y. Zhu y L. He, "Dual-fisheye lens stitching and error correction", en *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Shanghai: IEEE, oct. de 2017, págs. 1-6, ISBN: 9781538619377. DOI: 10.1109/CISP-BMEI.2017.8302053.

[154] H. T. Chen, B. Zhang, F. C. Sun, Y. L. Huang y J. Yuan, "Incremental scene detection in outdoor environment based on hierarchical bag-of-words model", *Kongzhi Lilun Yu Yingyong/Control Theory and Applications*, vol. 37, págs. 1471-1480, 7 jul. de 2020, ISSN: 10008152. DOI: 10.7641/CTA.2020.90683.

[155] S. Arshad y G.-W. Kim, "A Robust Feature Matching Strategy for Fast and Effective Visual Place Recognition in Challenging Environmental Conditions", *International Journal of Control, Automation and Systems*, vol. 21, págs. 948-962, 3 2023. DOI: 10.1007/s12555-021-0927-x.

[156] B. Wang, S. Wang, L. Ma y D. Qin, "A Robust and Efficient SLAM System in Dynamic Environment Based on Deep Features", vol. 854 LNEE, Springer Science y Business Media Deutschland GmbH, 2022, págs. 481-489, ISBN: 9789811694226. DOI: 10.1007/978-981-16-9423-3_60.

[157] J. Company-Corcoles, E. Garcia-Fidalgo y A. Ortiz, "Appearance-based loop closure detection combining lines and learned points for low-textured environments", *Autonomous Robots*, vol. 46, págs. 451-467, 3 2022. DOI: 10.1007/s10514-021-10032-7.

[158] S. Chen, B. Zhou, C. Jiang, W. Xue y Q. Li, "A lidar/visual slam backend with loop closure detection and graph optimization", *Remote Sensing*, vol. 13, 14 jul. de 2021, ISSN: 20724292. DOI: 10.3390/rs13142720.

[159] P. Loncomilla, J. R. del Solar y L. Martínez, "Object recognition using local invariant features for robotic applications: A survey", *Pattern Recognition*, vol. 60, págs. 499-514, dic. de 2016, ISSN: 0031-3203. DOI: 10.1016/J.PATCOG.2016.05.021.

[160] J. Wang e Y. Yagi, "Efficient topological localization using global and local feature matching", *International Journal of Advanced Robotic Systems*, vol. 10, ene. de 2013, ISSN: 17298806. DOI: 10.5772/55630/ASSET/IMAGES/LARGE/10.5772_55630-FIG6.JPEG. dirección: https://journals.sagepub.com/doi/10.5772/55630.

[161] L. Kabbai, M. Abdellaoui y A. Douik, "Image classification by combining local and global features", *Visual Computer*, vol. 35, págs. 679-693, 5 mayo de 2019, ISSN: 01782789. DOI: 10.1007/S00371-018-1503-0/TABLES/8. dirección: https://link.springer.com/article/10.1007/s00371-018-1503-0.

[162] B. F. Cura y E. Surer, "Scene classification: A comprehensive study combining local and global descriptors", Institute of Electrical y Electronics Engineers Inc., abr. de 2019, ISBN: 9781728119045. DOI: 10.1109/SIU.2019.8806590.

[163] Z. Mehmood, F. Abbas, T. Mahmood, M. A. Javid, A. Rehman y T. Nawaz, "Content-Based Image Retrieval Based on Visual Words Fusion Versus Features

Fusion of Local and Global Features", *Arabian Journal for Science and Engineering*, vol. 43, 12 2018, ISSN: 21914281. DOI: 10.1007/s13369-018-3062-0.

[164] Y. Su, S. Shan, X. Chen y W. Gao, "Hierarchical ensemble of global and local classifiers for face recognition", *Proceedings of the IEEE International Conference on Computer Vision*, 2007. DOI: 10.1109/ICCV.2007.4409060.

[165] N. Dalal y B. Triggs, "Histograms of oriented gradients for human detection", vol. I, 2005. DOI: 10.1109/CVPR.2005.177.

[166] A. Oliva y A. Torralba, *Building the gist of a scene: the role of global image features in recognition*, 2006. DOI: 10.1016/S0079-6123(06)55002-2.

[167] M. Rostkowska y P. Skrzypczyński, "Optimizing Appearance-Based Localization with Catadioptric Cameras: Small-Footprint Models for Real-Time Inference on Edge Devices", *Sensors*, vol. 23, n.º 14, 2023, ISSN: 1424-8220. DOI: 10.3390/s23146485. dirección: https://www.mdpi.com/1424-8220/23/14/6485.

[168] S. Wang, X. Lv, X. Liu y D. Ye, "Compressed Holistic ConvNet Representations for Detecting Loop Closures in Dynamic Environments", *IEEE Access*, vol. 8, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2982228.

[169] L. G. Camara y L. Přeučil, "Visual Place Recognition by spatial matching of high-level CNN features", *Robotics and Autonomous Systems*, vol. 133, 2020, ISSN: 09218890. DOI: 10.1016/j.robot.2020.103625.

[170] J. J. Cabrera, S. Cebollada, M. Flores, O. Reinoso y L. Payá, "Training, Optimization and Validation of a CNN for Room Retrieval and Description of Omnidirectional Images", *SN Computer Science*, vol. 3, n.º 4, pág. 271, jul. de 2022, ISSN: 2661-8907. DOI: 10.1007/s42979-022-01127-8. dirección: https://link.springer.com/10.1007/s42979-022-01127-8.

[171] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa y E. Romera, "Fusion and binarization of CNN features for robust topological localization across seasons", vol. 2016-November, 2016. DOI: 10.1109/IROS.2016.7759685.

[172] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft y M. Milford, "On the performance of ConvNet features for place recognition", vol. 2015-December, 2015. DOI: 10.1109/IROS.2015.7353986.

[173] K. Li, Y. Ma, X. Wang, L. Ji y N. Geng, "Evaluation of Global Descriptor Methods for Appearance-Based Visual Place Recognition", *Journal of Robotics*, vol. 2023, 2023, ISSN: 16879619. DOI: 10.1155/2023/9150357.

[174] A. Krizhevsky, I. Sutskever y G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", en *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012. dirección: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[175] Y. Uchida, "Local Feature Detectors, Descriptors, and Image Representations: A Survey", 2016.

[176] Y. Li, S. Wang, Q. Tian y X. Ding, "A survey of recent advances in visual feature detection", *Neurocomputing*, vol. 149, PB 2015, ISSN: 18728286. DOI: 10.1016/j.neucom.2014.08.003.

[177] J. Wang y W. Zhang, "A Survey of Corner Detection Methods", 2018. DOI: 10.2991/iceea-18.2018.47.

[178] J. Matas, O. Chum, M. Urban y T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions", vol. 22, 2004. DOI: 10.1016/j.imavis.2004.02.006.

[179] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, 6 1986, ISSN: 01628828. DOI: 10.1109/TPAMI.1986.4767851.

[180] W. Huang, Y. Wei, Y. Xie y H. Jin, "Survey of local invariant feature description", *Proceedings - 2013 Chinese Automation Congress, CAC 2013*, págs. 353-358, 2013. DOI: 10.1109/CAC.2013.6775758.

[181] M. Hassaballah, A. A. Abdelmgeid y H. A. Alshazly, "Image features detection, description and matching", *Studies in Computational Intelligence*, vol. 630, págs. 11-45, 2016, ISSN: 1860949X. DOI: 10.1007/978-3-319-28854-3_2/FIGURES/16. dirección: https://link.springer.com/chapter/10.1007/978-3-319-28854-3_2.

[182] K. Mikolajczyk y T. Tuytelaars, "Local Image Features", *Encyclopedia of Biometrics*, págs. 939-943, 2009. DOI: 10.1007/978-0-387-73003-5_224. dirección: https://link.springer.com/referenceworkentry/10.1007/978-0-387-73003-5_224.

[183] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, 2 2004, ISSN: 09205691. DOI: 10.1023/B:VISI.0000029664.99615.94.

[184] N. Savinov, A. Seki, L. Ladicky, T. Sattler y M. Pollefeys, "Quad-Networks: Unsupervised Learning to Rank for Interest Point Detection", en *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, jul. de 2017, págs. 3929-3937, ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.418.

[185] N. Zizakic y A. Pizurica, "Efficient Local Image Descriptors Learned with Autoencoders", *IEEE Access*, vol. 10, 2022, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3138168.

[186] B. G. Kumar, G. Carneiro e I. Reid, "Learning local image descriptors with deep siamese and triplet convolutional networks by minimizing global loss functions", vol. 2016-December, 2016. DOI: 10.1109/CVPR.2016.581.

[187] C. Osendorfer, J. Bayer, S. Urban y P. V. D. Smagt, "Convolutional Neural Networks learn compact local image descriptors", vol. 8228 LNCS, 2013, págs. 624-630, ISBN: 9783642420504. DOI: 10.1007/978-3-642-42051-1_77.

[188] Y. Tian, B. Fan y F. Wu, "L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space", en *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, págs. 6128-6136. DOI: 10.1109/CVPR.2017.649.

[189] Z. Luo et al., "GeoDesc: Learning local descriptors by integrating geometry constraints", vol. 11213 LNCS, 2018. DOI: 10.1007/978-3-030-01240-3_11.

[190] C. Liu, J. Xu y F. Wang, "A Review of Keypoints' Detection and Feature Description in Image Registration", *Scientific Programming*, vol. 2021, 2021, ISSN: 10589244. DOI: 10.1155/2021/8509164.

[191] C. Harris y M. Stephens, "A Combined Corner and Edge Detector", en *Procedings of the Alvey Vision Conference 1988*, Manchester: Alvey Vision Club, 1988, págs. 23.1-23.6. DOI: 10.5244/C.2.23. dirección: http://www.bmva.org/bmvc/1988/avc-88-023.html.

[192] S. Leutenegger, M. Chli y R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints", en *2011 International Conference on Computer Vision*, ISSN: 2380-7504, nov. de 2011, págs. 2548-2555. DOI: 10.1109/ICCV.2011.6126542. dirección: https://ieeexplore.ieee.org/document/6126542.

[193] Z. Wang, E. Simoncelli y A. Bovik, "Multiscale structural similarity for image quality assessment", en *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Pacific Grove, CA, USA: IEEE, 2003, págs. 1398-1402, ISBN: 9780780381049. DOI: 10.1109/ACSSC.2003.1292216. (visitado 18-11-2022).

[194] J. Ma, X. Jiang, A. Fan, J. Jiang y J. Yan, "Image Matching from Handcrafted to Deep Features: A Survey", *International Journal of Computer Vision*, vol. 129, págs. 23-79, 1 ene. de 2021, ISSN: 15731405. DOI: 10.1007/S11263-020-01359-2/TABLES/4. dirección: https://link.springer.com/article/10.1007/s11263-020-01359-2.

[195] D. DeTone, T. Malisiewicz y A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description", vol. 2018-June, 2018. DOI: 10.1109/CVPRW.2018.00060.

[196] P. H. Christiansen, M. F. Kragh, Y. Brodskiy y H. Karstoft, "UnsuperPoint: End-to-end Unsupervised Interest Point Detector and Descriptor", 2019. dirección: http://arxiv.org/abs/1907.04011.

[197] M. Dusmanu et al., "D2-Net: A Trainable CNN for Joint Detection and Description of Local Features", mayo de 2019. dirección: http://arxiv.org/abs/1905.03561.

[198] J. Revaud, P. Weinzaepfel, C. de Souza y M. Humenberger, "R2D2: Repeatable and reliable detector and descriptor", vol. 32, 2019.

[199] A. Konrad, C. Eising, G. Sistu, J. McDonald, R. Villing y S. Yogamani, "FisheyeSuperPoint: Keypoint Detection and Description Network for Fisheye Images:" en *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, SCITEPRESS - Science y Technology Publications, 2022, págs. 340-347, ISBN: 9789897585555. DOI: 10.5220/0010795400003124. dirección: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010795400003124.

[200] W. Tian, P. Cai, Y. Wen y X. Chu, "Robust Keypoint Detection and Matching on Fisheye Images by Self-Supervised Learning", 2022. DOI: 10.1155/2022/4024774. dirección: https://doi.org/10.1155/2022/4024774.

[201] K. Wilson y N. Snavely, "Robust Global Translations with 1DSfM", en *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele y T. Tuytelaars, eds., vol. 8691, Cham: Springer International Publishing, 2014, págs. 61-75, ISBN: 9783319105772 9783319105789. DOI: 10.1007/978-3-319-10578-9_5. dirección: http://link.springer.com/10.1007/978-3-319-10578-9_5.

[202] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context", en *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele y T. Tuytelaars, eds., Cham:

Springer International Publishing, 2014, págs. 740-755, ISBN: 978-3-319-10602-1.

[203] Z. Li y N. Snavely, "MegaDepth: Learning Single-View Depth Prediction from Internet Photos", en *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, jun. de 2018, págs. 2041-2050, ISBN: 9781538664209. DOI: 10.1109/CVPR.2018.00218. dirección: https://ieeexplore.ieee.org/document/8578316/.

[204] T. Sattler et al., "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions", en *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, jun. de 2018, págs. 8601-8610, ISBN: 9781538664209. DOI: 10.1109/CVPR.2018.00897. dirección: https://ieeexplore.ieee.org/document/8578995/.

[205] T. Sattler, T. Weyand, B. Leibe y L. Kobbelt, "Image Retrieval for Image-Based Localization Revisited", en *Procedings of the British Machine Vision Conference 2012*, Surrey: British Machine Vision Association, 2012, págs. 76.1-76.12, ISBN: 9781901725469. DOI: 10.5244/C.26.76. dirección: http://www.bmva.org/bmvc/2012/BMVC/paper076/index.html.

[206] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis y O. Chum, "Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking", en *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, jun. de 2018, págs. 5706-5715, ISBN: 9781538664209. DOI: 10.1109/CVPR.2018.00598. dirección: https://ieeexplore.ieee.org/document/8578696/.

[207] W. Maddern, G. Pascoe, C. Linegar y P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset", *International Journal of Robotics Research*, vol. 36, 1 2017, ISSN: 17413176. DOI: 10.1177/0278364916679498.

[208] S. Yogamani et al., "WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving", en *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, págs. 9307-9317. DOI: 10.1109/ICCV.2019.00940.

[209] M. Lourenco, J. P. Barreto y F. Vasconcelos, "SRD-SIFT: Keypoint detection and matching in images with radial distortion", *IEEE Transactions on Robotics*, vol. 28, 3 2012, ISSN: 15523098. DOI: 10.1109/TRO.2012.2184952.

[210] Y. Zhang, J. Song, Y. Ding, Y. Yuan y H. L. Wei, "FSD-BRIEF: A Distorted BRIEF Descriptor for Fisheye Image Based on Spherical Perspective Model", *Sensors 2021, Vol. 21, Page 1839*, vol. 21, pág. 1839, 5 mar. de 2021, ISSN: 1424-8220. DOI: 10.3390/S21051839. dirección: https://www.mdpi.com/1424-8220/21/5/1839/htmhttps://www.mdpi.com/1424-8220/21/5/1839.

[211] M. P. de Melo, L. Cambuim y E. Barros, "Occupancy Grid Map Estimation Based on Visual SLAM and Ground Segmentation", en *2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*, 2021, págs. 288-293. DOI: 10.1109/LARS/SBR/WRE54079.2021.9605417.

[212] R. C. Luo y W. Shih, "Autonomous Mobile Robot Intrinsic Navigation Based on Visual Topological Map", en *2018 IEEE 27th International Symposium on*

Industrial Electronics (ISIE), 2018, págs. 541-546. DOI: 10.1109/ISIE.2018.8433588.

[213] E. Garcia-Fidalgo y A. Ortiz, "Hierarchical Place Recognition for Topological Mapping", *IEEE Transactions on Robotics*, vol. 33, n.º 5, págs. 1061-1074, 2017. DOI: 10.1109/TRO.2017.2704598.

[214] E. Garcia-Fidalgo y A. Ortiz, "Vision-based topological mapping and localization methods: A survey", *Robotics and Autonomous Systems*, vol. 64, págs. 1-20, feb. de 2015, ISSN: 0921-8890. DOI: 10.1016/J.ROBOT.2014.11.009.

[215] H. Badino, D. Huber y T. Kanade, "Visual topometric localization", en *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, págs. 794-799. DOI: 10.1109/IVS.2011.5940504.

[216] A. G. Ozkil et al., "Practical indoor mobile robot navigation using hybrid maps", en *2011 IEEE International Conference on Mechatronics*, 2011, págs. 475-480. DOI: 10.1109/ICMECH.2011.5971333.

[217] L. Payá, A. Peidró, F. Amorós, D. Valiente y O. Reinoso, "Modeling Environments Hierarchically with Omnidirectional Imaging and Global-Appearance Descriptors", *Remote Sensing*, vol. 10, n.º 4, 2018, ISSN: 2072-4292. DOI: 10.3390/rs10040522. dirección: https://www.mdpi.com/2072-4292/10/4/522.

[218] X. Qi et al., "Building semantic grid maps for domestic robot navigation", *International Journal of Advanced Robotic Systems*, vol. 17, 1 2020, ISSN: 17298814. DOI: 10.1177/1729881419900066.

[219] P. E. Sarlin, C. Cadena, R. Siegwart y M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale", vol. 2019-June, 2019. DOI: 10.1109/CVPR.2019.01300.

[220] V. Román, L. Payá, S. Cebollada, A. Peidró y óscar Reinoso, "Evaluating the Robustness of New Holistic Description Methods in Position Estimation of Mobile Robots", vol. 793, 2022. DOI: 10.1007/978-3-030-92442-3_12.

[221] T. Zhang, S. Guo, L. Ma y W. Meng, "Hierarchical Image Retrieval Method Based on Bag-of-Visual-Word and Eight-point Algorithm with Feature Clouds for Visual Indoor Positioning", 2022. DOI: 10.1109/APCC55198.2022.9943709.

[222] Q. Zhou, T. Sattler, M. Pollefeys y L. Leal-Taixe, "To Learn or Not to Learn: Visual Localization from Essential Matrices", en *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France: IEEE, mayo de 2020, págs. 3319-3326, ISBN: 9781728173955. DOI: 10.1109/ICRA40945.2020.9196607. dirección: https://ieeexplore.ieee.org/document/9196607/.

[223] J. Jiang, Y. Zou, L. Chen e Y. Fang, "A visual and vae based hierarchical indoor localization method", *Sensors*, vol. 21, 10 2021. DOI: 10.3390/s21103406.

[224] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi y T. Pajdla, "24/7 place recognition by view synthesis", vol. 07-12-June-2015, 2015. DOI: 10.1109/CVPR.2015.7298790.

[225] D. Nister, "An efficient solution to the five-point relative pose problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, 6 2004, ISSN: 01628828. DOI: 10.1109/TPAMI.2004.17.

[226] M. Humenberger et al., "Investigating the Role of Image Retrieval for Visual Localization: An Exhaustive Benchmark", *International Journal of Computer*

Vision, vol. 130, 7 2022, ISSN: 15731405. DOI: 10.1007/s11263-022-01615-7.

[227] L. Fernández, L. Payá, O. Reinoso, L. M. Jiménez y M. Ballesta, "A Study of Visual Descriptors for Outdoor Navigation Using Google Street View Images", *Journal of Sensors*, vol. 2016, 2016, ISSN: 16877268. DOI: 10.1155/2016/1537891.

[228] S. A. Mohamed, M. H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen y J. Plosila, "A Survey on Odometry for Autonomous Navigation Systems", *IEEE Access*, vol. 7, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2929133.

[229] Y. Alkendi, L. Seneviratne e Y. Zweiri, "State of the Art in Vision-Based Localization Techniques for Autonomous Navigation Systems", *IEEE Access*, vol. 9, págs. 76 847-76 874, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3082778.

[230] A. Kendall, M. Grimes y R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization", en *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, págs. 2938-2946. DOI: 10.1109/ICCV.2015.336.

[231] J. Engel, J. Stückler y D. Cremers, "Large-scale direct SLAM with stereo cameras", vol. 2015-December, 2015. DOI: 10.1109/IROS.2015.7353631.

[232] P. Liu, L. Heng, T. Sattler, A. Geiger y M. Pollefeys, "Direct visual odometry for a fisheye-stereo camera", *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-September, págs. 1746-1752, dic. de 2017, ISSN: 21530866. DOI: 10.1109/IROS.2017.8205988.

[233] H. Seok y J. Lim, "ROVO: Robust omnidirectional visual odometry for wide-baseline wide-FOV camera systems", *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, págs. 6344-6350, mayo de 2019, ISSN: 10504729. DOI: 10.1109/ICRA.2019.8793758.

[234] Z. Javed y G. W. Kim, "OmniVO: Towards Robust Omni Directional Visual Odometry with Multi-Camera Collaboration for Challenging Conditions", *IEEE Access*, 2022, ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3204870.

[235] H. Matsuki, L. V. Stumberg, V. Usenko, J. Stuckler y D. Cremers, "Omnidirectional DSO: Direct Sparse Odometry with Fisheye Cameras", *IEEE Robotics and Automation Letters*, vol. 3, págs. 3693-3700, 4 oct. de 2018, ISSN: 23773766. DOI: 10.1109/LRA.2018.2855443.

[236] L. R. Agostinho, N. M. Ricardo, M. I. Pereira, A. Hiolle y A. M. Pinto, "A Practical Survey on Visual Odometry for Autonomous Driving in Challenging Scenarios and Conditions", *IEEE Access*, vol. 10, págs. 72 182-72 205, 2022. DOI: 10.1109/ACCESS.2022.3188990.

[237] L. R. Agostinho, N. M. Ricardo, M. I. Pereira, A. Hiolle y A. M. Pinto, "A Practical Survey on Visual Odometry for Autonomous Driving in Challenging Scenarios and Conditions", *IEEE Access*, vol. 10, 2022, ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3188990.

[238] D. Caruso, J. Engel y D. Cremers, "Large-scale direct SLAM for omnidirectional cameras", vol. 2015-December, 2015. DOI: 10.1109/IROS.2015.7353366.

[239] D. Scaramuzza y F. Fraundorfer, "Tutorial: Visual odometry", *IEEE Robotics and Automation Magazine*, vol. 18, 4 2011, ISSN: 10709932. DOI: 10.1109/MRA.2011.943233.

[240] O. Faugeras y F. Lustman, "Motion and structure from motion in a piecewise planar environment", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 02, n.º 03, págs. 485-508, sep. de 1988, ISSN: 0218-0014, 1793-6381. DOI: 10.1142/S0218001488000285. dirección: https://www.worldscientific.com/doi/abs/10.1142/S0218001488000285.

[241] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel y J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM", *IEEE Transactions on Robotics*, vol. 37, págs. 1874-1890, 6 dic. de 2021, ISSN: 1552-3098. DOI: 10.1109/TRO.2021.3075644. dirección: https://ieeexplore.ieee.org/document/9440682/.

[242] R. Mur-Artal y J. D. Tardos, "Visual-Inertial Monocular SLAM with Map Reuse", *IEEE Robotics and Automation Letters*, vol. 2, 2 2017, ISSN: 23773766. DOI: 10.1109/LRA.2017.2653359.

[243] S. Wang, R. Clark, H. Wen y N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks", 2017. DOI: 10.1109/ICRA.2017.7989236.

[244] R. Li, S. Wang, Z. Long y D. Gu, "UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning", 2018. DOI: 10.1109/ICRA.2018.8461251.

[245] T. Pandey, D. Pena, J. Byrne y D. Moloney, "Leveraging deep learning for visual odometry using optical flow", *Sensors*, vol. 21, 4 2021, ISSN: 14248220. DOI: 10.3390/s21041313.

[246] S. Baker y S. K. Nayar, "Theory of single-viewpoint catadioptric image formation", *International Journal of Computer Vision*, vol. 35, 2 1999, ISSN: 09205691. DOI: 10.1023/A:1008128724364.

[247] X. Ying y Z. Hu, "Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model", en *Computer Vision - ECCV 2004*, T. Pajdla y J. Matas, eds., ép. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2004, págs. 442-455, ISBN: 9783540246701. DOI: 10.1007/978-3-540-24670-1_34.

[248] S. Ramalingam, "Field of View", en *Computer Vision: A Reference Guide*, K. Ikeuchi, ed. Boston, MA: Springer US, 2014, págs. 294-297, ISBN: 978-0-387-31439-6. DOI: 10.1007/978-0-387-31439-6_462. dirección: https://doi.org/10.1007/978-0-387-31439-6_462.

[249] L. Puig, J. Bermúdez, P. Sturm y J. Guerrero, "Calibration of omnidirectional cameras in practice: A comparison of methods", *Computer Vision and Image Understanding*, vol. 116, n.º 1, págs. 120-137, 2012, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2011.08.003. dirección: https://www.sciencedirect.com/science/article/pii/S1077314211001858.

[250] B. Micusik y T. Pajdla, "Structure from motion with wide circular field of view cameras", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, 7 2006, ISSN: 01628828. DOI: 10.1109/TPAMI.2006.151.

[251] J. P. Barreto y H. Araujo, "Geometric properties of central catadioptric line images and their application in calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, 8 2005, ISSN: 01628828. DOI: 10.1109/TPAMI.2005.163.

[252] L. Puig, Y. Bastanlar, P. Sturm, J. J. Guerrero y J. Barreto, "Calibration of central catadioptric cameras using a DLT-like approach", *International Journal of Computer Vision*, vol. 93, 1 2011, ISSN: 15731405. DOI: 10.1007/s11263-010-0411-1.

[253] J. Fan, J. Zhang, S. J. Maybank y D. Tao, "Wide-angle image rectification: a survey", *International Journal of Computer Vision*, vol. 130, n.º 3, págs. 747-776, mar. de 2022, ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-021-01562-9. dirección: https://link.springer.com/10.1007/s11263-021-01562-9.

[254] V. R. Kumar, C. Eising, C. Witt y S. K. Yogamani, "Surround-View Fisheye Camera Perception for Automated Driving: Overview, Survey & Challenges", *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, n.º 4, págs. 3638-3659, 2023. DOI: 10.1109/TITS.2023.3235057.

[255] C. Geyer y K. Daniilidis, "A Unifying Theory for Central Panoramic Systems and Practical Implications", en *Computer Vision — ECCV 2000*, D. Vernon, ed., vol. 1843, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, págs. 445-461, ISBN: 9783540676867 9783540450535. DOI: 10.1007/3-540-45053-X_29. dirección: http://link.springer.com/10.1007/3-540-45053-X_29.

[256] J. Barreto y H. Araujo, "Issues on the geometry of central catadioptric image formation", en *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2, 2001, págs. II-II. DOI: 10.1109/CVPR.2001.990992.

[257] J. Courbon, Y. Mezouar y P. Martinet, "Evaluation of the Unified Model of the Sphere for Fisheye Cameras in Robotic Applications", *Advanced Robotics*, vol. 26, n.º 8-9, págs. 947-967, mayo de 2012, ISSN: 0169-1864, 1568-5535. DOI: 10.1163/156855312X633057. dirección: https://www.tandfonline.com/doi/full/10.1163/156855312X633057.

[258] J. Courbon, Y. Mezouar, L. Eckt y P. Martinet, "A generic fisheye camera model for robotic applications", en *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA: IEEE, oct. de 2007, págs. 1683-1688, ISBN: 9781424409112. DOI: 10.1109/IROS.2007.4399233. dirección: https://ieeexplore.ieee.org/document/4399233/.

[259] R. Tezaur, A. Kumar y O. Nestares, "A New Non-central Model for Fisheye Calibration", en *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, págs. 5218-5227. DOI: 10.1109/CVPRW56347.2022.00570.

[260] O. Bogdan, V. Eckstein, F. Rameau y J.-C. Bazin, "DeepCalib: A Deep Learning Approach for Automatic Intrinsic Calibration of Wide Field-of-View Cameras", en *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production*, ép. CVMP '18, London, United Kingdom: Association for Computing Machinery, 2018, ISBN: 9781450360586. DOI: 10.1145/3278471.3278479. dirección: https://doi.org/10.1145/3278471.3278479.

[261] N. Wakai y T. Yamashita, "Deep Single Fisheye Image Camera Calibration for Over 180-degree Projection of Field of View", en *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, págs. 1174-1183. DOI: 10.1109/ICCVW54120.2021.00137.

[262] Garmin, *VIRB 360*, es-ES. dirección: https://www.garmin.com/es-ES/p/562010 (visitado 18-11-2022).

[263] *Davide Scaramuzza - OCamCalib: Omnidirectional Camera Calibration Toolbox for Matlab*, es-419. dirección: https://sites.google.com/site/scarabotix/ocamcalib-omnidirectional-camera-calibration-toolbox-for-matlab (visitado 21-09-2023).

[264] I. Kostavelis, K. Charalampous, A. Gasteratos y J. K. Tsotsos, "Robot navigation via spatial and temporal coherent semantic maps", *Engineering Applications of Artificial Intelligence*, vol. 48, 2016, ISSN: 09521976. DOI: 10.1016/j.engappai.2015.11.004.

[265] S. Harapanahalli, N. O. Mahony, G. V. Hernandez, S. Campbell, D. Riordan y J. Walsh, "Autonomous navigation of mobile robots in factory environment", vol. 38, 2019. DOI: 10.1016/j.promfg.2020.01.134.

[266] C. Patruno, R. Colella, M. Nitti, V. Renò, N. Mosca y E. Stella, "A vision-based odometer for localization of omnidirectional indoor robots", *Sensors*, vol. 20, 3 2020, ISSN: 14248220. DOI: 10.3390/s20030875.

[267] H. Taheri y Z. C. Xia, "SLAM; definition and evolution", *Engineering Applications of Artificial Intelligence*, vol. 97, pág. 104 032, ene. de 2021, ISSN: 0952-1976. DOI: 10.1016/J.ENGAPPAI.2020.104032.

[268] M. B. Alatise y G. P. Hancke, "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods", *IEEE Access*, vol. 8, págs. 39 830-39 846, 2020.

[269] Y. Jiang, Y. Xu e Y. Liu, "Performance evaluation of feature detection and matching in stereo visual odometry", *Neurocomputing*, vol. 120, págs. 380-390, 2013, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2012.06.055.

[270] F. Andert y L. Goormann, "Combined grid and feature-based occupancy map building in large outdoor environments", en *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, págs. 2065-2070. DOI: 10.1109/IROS.2007.4399086.

[271] Y. Liu, J. Chen y X. Bai, "An Approach for Multi-Objective Obstacle Avoidance Using Dynamic Occupancy Grid Map", en *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2020, págs. 1209-1215. DOI: 10.1109/ICMA49215.2020.9233760.

[272] V. Román, L. Payá, S. Cebollada y Ó. Reinoso, "Creating Incremental Models of Indoor Environments through Omnidirectional Imaging", *Applied Sciences*, vol. 10, n.º 18, 2020, ISSN: 2076-3417. DOI: 10.3390/app10186480.

[273] M. Yuan, W. Yau y Z. Li, "Lost Robot Self-Recovery via Exploration Using Hybrid Topological-Metric Maps", en *TENCON 2018 - 2018 IEEE Region 10 Conference*, 2018, págs. 0188-0193. DOI: 10.1109/TENCON.2018.8650236.

[274] A. Gil, M. Juliá y O. Reinoso, "Occupancy grid based graph-SLAM using the distance transform, SURF features and SGD", *Engineering Applications of Ar-*

tificial Intelligence, vol. 40, págs. 1-10, 2015, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2014.12.010.

[275] S. T. O'Callaghan y F. T. Ramos, "Gaussian process occupancy maps", *The International Journal of Robotics Research*, vol. 31, n.º 1, págs. 42-62, 2012.

[276] M. G. Jadidi, L. Gan, S. A. Parkison, J. Li y R. M. Eustice, "Gaussian processes semantic map representation", *arXiv preprint arXiv:1707.01532*, 2017.

[277] X. Song, Z. Cao y H. Gao, "Local Gaussian Processes for Identifying Complex Mobile robot System", en *2018 37th Chinese Control Conference (CCC)*, 2018, págs. 3796-3802.

[278] K. Polymenakos et al., "Safety guarantees for planning based on iterative Gaussian processes", *arXiv preprint arXiv:1912.00071*, 2019.

[279] K. Sun, K. Saulnier, N. Atanasov, G. J. Pappas y V. Kumar, "Dense 3-d mapping with spatial correlation via gaussian filtering", en *2018 Annual American Control Conference (ACC)*, IEEE, 2018, págs. 4267-4274.

[280] S. Park, Y. Huang, C. F. Goh y K. Shimada, "Robot Model Learning with Gaussian Process Mixture Model", en *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, págs. 1263-1268.

[281] A. Dalla-Libera, E. Tosello, G. Pillonetto, S. Ghidoni y R. Carli, "Proprioceptive Robot Collision Detection through Gaussian Process Regression", en *2019 American Control Conference (ACC)*, 2019, págs. 19-24.

[282] M. K. Nutalapati, L. Arora, A. Bose, K. Rajawat y R. M. Hegde, "Model Free Calibration of Wheeled Robots Using Gaussian Process", en *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, págs. 29-35.

[283] B. Li, Y. Wang, Y. Zhang, W. Zhao, J. Ruan y P. Li, "GP-SLAM: laser-based SLAM approach based on regionalized Gaussian process map reconstruction", *Autonomous Robots*, págs. 1-21, 2020.

[284] L. Nguyen, J. V. Miro, L. Shi y T. Vidal-Calleja, "Gaussian Mixture Marginal Distributions for Modelling Remaining Metallic Pipe Wall Thickness", en *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2019, págs. 257-262. DOI: 10.1109/CIS-RAM47153.2019.9095851.

[285] O. Reinoso y L. Payá, "Special Issue on Visual Sensors", *Sensors*, vol. 20, n.º 3, 2020, ISSN: 1424-8220. DOI: 10.3390/s20030910.

[286] S. Emani, K. P. Soman, V. V. Sajith Variyar y S. Adarsh, "Obstacle Detection and Distance Estimation for Autonomous Electric Vehicle Using Stereo Vision and DNN", en *Soft Computing and Signal Processing*, J. Wang, G. R. M. Reddy, V. K. Prasad y V. S. Reddy, eds., Singapore: Springer Singapore, 2019, págs. 639-648, ISBN: 978-981-13-3393-4.

[287] H. Zhang, D. E. Hernandez, Z. Su y B. Su, "A Low Cost Vision-Based Road-Following System for Mobile Robots", *Applied Sciences*, vol. 8, n.º 9, 2018, ISSN: 2076-3417. DOI: 10.3390/app8091635.

[288] S. Jung, U. Lee, J. Jung y D. H. Shim, "Real-time Traffic Sign Recognition system with deep convolutional neural network", en *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2016, págs. 31-34.

[289] F. Amorós, L. Payá, W. Mayol-Cuevas, L. M. Jiménez y O. Reinoso, "Holistic Descriptors of Omnidirectional Color Images and Their Performance in Estimation of Position and Orientation", *IEEE Access*, vol. 8, págs. 81 822-81 848, 2020. DOI: 10.1109/ACCESS.2020.2990996.

[290] L. Payá, A. Gil y O. Reinoso, "A state-of-the-art review on mapping and localization of mobile robots using omnidirectional vision sensors", *Journal of Sensors*, vol. 2017, 2017.

[291] S. Li, "Full-View Spherical Image Camera", en *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4, 2006, págs. 386-390. DOI: 10.1109/ICPR.2006.585.

[292] S. Barone, P. Neri, A. Paoli y A. Razionale, "Catadioptric stereo-vision system using a spherical mirror", *Procedia Structural Integrity*, vol. 8, págs. 83 -91, 2018, AIAS2017 - 46th Conference on Stress Analysis and Mechanical Engineering Design, 6-9 September 2017, Pisa, Italy, ISSN: 2452-3216. DOI: https://doi.org/10.1016/j.prostr.2017.12.010.

[293] J. M. Junior, A. Tommaselli y M. Moraes, "Calibration of a catadioptric omnidirectional vision system with conic mirror", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 113, págs. 97 -105, 2016, ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2015.10.008.

[294] R. Boutteau, X. Savatier, J.-Y. Ertaud y B. Mazari, "A 3D Omnidirectional Sensor For Mobile Robot Applications", en *Mobile Robots Navigation*, mar. de 2010.

[295] W. Gao, K. Wang, W. Ding, F. Gao, T. Qin y S. Shen, "Autonomous aerial robot using dual-fisheye cameras", *J. Field Robotics*, vol. 37, págs. 497-514, 2020.

[296] G. H. Lee, F. Fraundorfer y M. Pollefeys, "Structureless pose-graph loop-closure with a multi-camera system on a self-driving car", en *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, págs. 564-571. DOI: 10.1109/IROS.2013.6696407.

[297] S. Poddar, R. Kottath y V. Karar, "Evolution of visual odometry techniques", *arXiv preprint arXiv:1804.11142*, 2018.

[298] D. Valiente, L. Fernández, A. Gil, L. Payá y O. Reinoso, "Visual odometry through appearance-and feature-based method with omnidirectional images", *Journal of Robotics*, vol. 2012, 2012.

[299] Q. Xiao, X. Liu y M. Liu, "Object Tracking Based on Local Feature Matching", en *2012 Fifth International Symposium on Computational Intelligence and Design*, vol. 1, 2012, págs. 399-402. DOI: 10.1109/ISCID.2012.106.

[300] A. Jakubović y J. Velagić, "Image Feature Matching and Object Detection Using Brute-Force Matchers", en *2018 International Symposium ELMAR*, 2018, págs. 83-86. DOI: 10.23919/ELMAR.2018.8534641.

[301] Z. Zivkovic, B. Bakker y B. Krose, "Hierarchical map building using visual landmarks and geometric constraints", en *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, págs. 2480-2485. DOI: 10.1109/IROS.2005.1544951.

[302] H. Hu, B.-F. Wu, W.-C. Lu y C.-L. Jen, "Monocular Vision-Based Robot Localization and Target Tracking", *Journal of Robotics*, vol. 2011, pág. 548 042,

2011. DOI: 10.1155/2011/548042. dirección: https://doi.org/10.1155/2011/548042.

[303] R. Hartley y A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[304] D. Nister, "An efficient solution to the five-point relative pose problem", en *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2, 2003, págs. II-195. DOI: 10.1109/CVPR.2003.1211470.

[305] D. Scaramuzza, "1-Point-RANSAC Structure from Motion for Vehicle-Mounted Cameras by Exploiting Non-holonomic Constraints", *International Journal of Computer Vision*, vol. 95, n.º 1, págs. 74-85, 2011. DOI: 10.1007/s11263-011-0441-3.

[306] Robotics and Perception Group, University of Zurich., *The "Multi-FoV"synthetic datasets. Omnidirectional image dataset*, accessed January 14, 2021, 2013. dirección: http://rpg.ifi.uzh.ch/fov.html/.

[307] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections", *Nature*, vol. 293, n.º 5828, págs. 133-135, 1981.

[308] S. Thrun, W. Burgard y D. Fox, *Probabilistic Robotics* (Intelligent Robotics and Autonomous Agents series). MIT Press, 2005, ISBN: 9780262303804.

[309] R. I. Hartley y P. Sturm, "Triangulation", *Computer Vision and Image Understanding*, vol. 68, n.º 2, págs. 146 -157, 1997, ISSN: 1077-3142. DOI: https://doi.org/10.1006/cviu.1997.0547.

[310] D. Scaramuzza, A. Martinelli y R. Siegwart, "A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion", en *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, 2006a, págs. 45-45. DOI: 10.1109/ICVS.2006.3.

[311] N. S. Nair y M. S. Nair, "On evolutionary computation techniques for multi-view triangulation", *Machine Vision and Applications*, vol. 31, pág. 29, 2020.

[312] P. A. Beardsley, A. Zisserman y D. W. Murray, "Navigation using affine structure from motion", en *European Conference on Computer Vision*, Springer, 1994, págs. 85-96.

[313] M. G. Jadidi, J. V. Miro y G. Dissanayake, "Gaussian processes autonomous mapping and exploration for range-sensing mobile robots", *Autonomous Robots*, vol. 42, n.º 2, págs. 273-290, 2018.

[314] V. Tresp, "A Bayesian committee machine", *Neural computation*, vol. 12, n.º 11, págs. 2719-2741, 2000.

[315] T. Cover y P. Hart, "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, vol. 13, n.º 1, págs. 21-27, 1967.

[316] K. Chomboon, P. Chujai, P. Teerarassamee, K. Kerdprasop y N. Kerdprasop, "An empirical study of distance metrics for k-nearest neighbor algorithm", en *Proceedings of the 3rd international conference on industrial application engineering*, 2015, págs. 280-285.

[317] J. L. Bentley, "Multidimensional binary search trees used for associative searching", *Communications of the ACM*, vol. 18, n.º 9, págs. 509-517, 1975.

[318] K. Mikolajczyk y C. Schmid, "A performance evaluation of local descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, n.º 10, págs. 1615-1630, 2005.

[319] M. Hassaballah y K. M. Hosny, *Recent Advances in Computer Vision: Theories and Applications*. Springer, 2018, vol. 804.

[320] K. Yan, *RANSAC algorithm with example of finding homography*, MATLAB Central File Exchange. Retrieved October 1, 2021, MATLAB Central File Exchange, 2011. dirección: https://www.mathworks.com/matlabcentral/fileexchange/30809-ransac-algorithm-with-example-of-finding-homography.

[321] Z. Zhang, H. Rebecq, C. Forster y D. Scaramuzza, "Benefit of large field-of-view cameras for visual odometry", en *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, págs. 801-808.

[322] G. Yu y J.-M. Morel, "ASIFT: An Algorithm for Fully Affine Invariant Comparison", *Image Processing On Line*, vol. 1, págs. 11-38, 2011.

[323] S. Delmas et al., "SpheriCol: A Driving Assistance System for Power Wheelchairs Based on Spherical Vision and Range Measurements", en *2021 IEEE/SICE International Symposium on System Integration (SII)*, Iwaki, Fukushima, Japan: IEEE, ene. de 2021, págs. 505-510, ISBN: 9781728176581. DOI: 10.1109/IEEECONF49454.2021.9382766.

[324] V. K. L. Ha, R. Chai y H. T. Nguyen, "A Telepresence Wheelchair with 360-Degree Vision Using WebRTC", *Applied Sciences*, vol. 10, n.º 1, pág. 369, ene. de 2020, ISSN: 2076-3417. DOI: 10.3390/app10010369. dirección: https://www.mdpi.com/2076-3417/10/1/369.

[325] F. Morbidi et al., "Assistive Robotic Technologies for Next-Generation Smart Wheelchairs: Codesign and Modularity to Improve Users' Quality of Life", *IEEE Robotics & Automation Magazine*, págs. 2-14, 2022, ISSN: 1070-9932, 1558-223X. DOI: 10.1109/MRA.2022.3178965. dirección: https://ieeexplore.ieee.org/document/9806056/.

[326] J. Zhang, X. Yin, J. Luan y T. Liu, "An improved vehicle panoramic image generation algorithm", *Multimedia Tools and Applications*, vol. 78, n.º 19, págs. 27 663-27 682, oct. de 2019, ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-019-07890-w.

[327] S. Cebollada, L. Payá, M. Flores, V. Román, A. Peidró y O. Reinoso, "A Localization Approach Based on Omnidirectional Vision and Deep Learning", en *Informatics in Control, Automation and Robotics*, O. Gusikhin, K. Madani y J. Zaytoon, eds., Cham: Springer International Publishing, 2022, págs. 226-246, ISBN: 9783030924423. DOI: 10.1007/978-3-030-92442-3_13.

[328] V. Román, L. Payá, A. Peidró, M. Ballesta y O. Reinoso, "The Role of Global Appearance of Omnidirectional Images in Relative Distance and Orientation Retrieval", *Sensors*, vol. 21, n.º 10, pág. 3327, mayo de 2021, ISSN: 1424-8220. DOI: 10.3390/s21103327.

[329] S. Cebollada, L. Payá, X. Jiang y O. Reinoso, "Development and use of a convolutional neural network for hierarchical appearance-based localization", *Artificial Intelligence Review*, sep. de 2021, ISSN: 0269-2821, 1573-7462. DOI:

10.1007/s10462-021-10076-2. dirección: https://link.springer.com/10.1007/s10462-021-10076-2.

[330] A. Rana, C. Ozcinar y A. Smolic, "Towards generating ambisonics using audio-visual cue for virtual reality", en *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom: IEEE, mayo de 2019, págs. 2012-2016, ISBN: 9781479981311. DOI: 10.1109/ICASSP.2019.8683318.

[331] D. Gledhill, G. Y. Tian, D. Taylor y D. Clarke, "Panoramic imaging—a review", *Computers & Graphics*, vol. 27, n.º 3, págs. 435-445, jun. de 2003, ISSN: 00978493. DOI: 10.1016/S0097-8493(03)00038-4.

[332] Samsung, *Gear 360 (2017) | Samsung Soporte España*, es-ES. dirección: https://www.samsung.com/es/support/model/SM-R210NZWAPHE/ (visitado 18-11-2022).

[333] Ricoh, *Producto | RICOH THETA S*. dirección: https://theta360.com/es/about/theta/s.html (visitado 18-11-2022).

[334] S. Colonnese, F. Cuomo, L. Ferranti y T. Melodia, "Efficient video streaming of 360° cameras in Unmanned Aerial Vehicles: an analysis of real video sources", en *2018 7th European Workshop on Visual Information Processing (EUVIP)*, 2018, págs. 1-6. DOI: 10.1109/EUVIP.2018.8611639.

[335] H.-E. Benseddik, F. Morbidi y G. Caron, "PanoraMIS: An ultra-wide field of view image dataset for vision-based robot-motion estimation", *The International Journal of Robotics Research*, vol. 39, n.º 9, págs. 1037-1051, ago. de 2020, ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364920915248.

[336] Y. Zhang y F. Huang, "Panoramic visual slam technology for spherical images", *Sensors*, vol. 21, n.º 3, pág. 705, ene. de 2021, ISSN: 1424-8220. DOI: 10.3390/s21030705.

[337] W. Lyu, Z. Zhou, L. Chen e Y. Zhou, "A survey on image and video stitching", *Virtual Reality & Intelligent Hardware*, vol. 1, n.º 1, págs. 55-83, feb. de 2019, ISSN: 20965796. DOI: 10.3724/SP.J.2096-5796.2018.0008.

[338] S.-H. Lee y S.-J. Lee, "Development of remote automatic panorama VR imaging rig systems using smartphones", *Cluster Computing*, vol. 21, n.º 1, págs. 1175-1185, mar. de 2018, ISSN: 1386-7857, 1573-7543. DOI: 10.1007/s10586-017-0930-4.

[339] W. Zhang, Y. Wang e Y. Liu, "Generating High-Quality Panorama by View Synthesis Based on Optical Flow Estimation", *Sensors*, vol. 22, n.º 2, pág. 470, ene. de 2022, ISSN: 1424-8220. DOI: 10.3390/s22020470.

[340] M. Flores, D. Valiente, A. Gil, O. Reinoso y L. Payá, "Efficient probability-oriented feature matching using wide field-of-view imaging", *Engineering Applications of Artificial Intelligence*, vol. 107, pág. 104 539, ene. de 2022, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2021.104539.

[341] T. Wang, Y.-Y. Hsieh, F.-W. Wong e Y.-F. Chen, "Mask-RCNN Based People Detection Using A Top-View Fisheye Camera", en *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, Kaohsiung, Taiwan: IEEE, nov. de 2019, págs. 1-4, ISBN: 9781728146669. DOI: 10.1109/TAAI48200.2019.8959887.

[342] C. Tian, X. Chai y F. Shao, "Stitched image quality assessment based on local measurement errors and global statistical properties", *Journal of Visual Communication and Image Representation*, vol. 81, pág. 103 324, nov. de 2021, ISSN: 10473203. DOI: 10.1016/j.jvcir.2021.103324.

[343] I.-C. Lo, K.-T. Shih y H. H. Chen, "Image Stitching for Dual Fisheye Cameras", en *2018 25th IEEE International Conference on Image Processing (ICIP)*, Athens: IEEE, oct. de 2018, págs. 3164-3168, ISBN: 9781479970612. DOI: 10.1109/ICIP.2018.8451333.

[344] L. Xue, J. Zhu, H. Zhang y R. Liu, "A high-quality stitching algorithm based on fisheye images", *Optik*, vol. 238, pág. 166 520, jul. de 2021, ISSN: 00304026. DOI: 10.1016/j.ijleo.2021.166520.

[345] I.-C. Lo, K.-T. Shih y H. H. Chen, "Efficient and Accurate Stitching for 360° Dual-Fisheye Images and Videos", *IEEE Transactions on Image Processing*, vol. 31, págs. 251-262, 2022, ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2021.3130531.

[346] B.-H. Lin, H.-Z. Cheng, Y.-T. Li y J.-I. Guo, "360 Degree Fish Eye Optical Construction For Equirectangular Projection of Panoramic Images", en *2020 International Conference on Pervasive Artificial Intelligence (ICPAI)*, Taipei, Taiwan: IEEE, dic. de 2020, págs. 194-198, ISBN: 9781665404839. DOI: 10.1109/ICPAI51961.2020.00043.

[347] S.-H. Chou, C. Sun, W.-Y. Chang, W.-T. Hsu, M. Sun y J. Fu, "360-Indoor: Towards Learning Real-World Objects in 360° Indoor Equirectangular Images", en *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass Village, CO, USA: IEEE, mar. de 2020, págs. 834-842, ISBN: 9781728165530. DOI: 10.1109/WACV45572.2020.9093262. dirección: https://ieeexplore.ieee.org/document/9093262/ (visitado 26-10-2023).

[348] J. Xiao, K. A. Ehinger, A. Oliva y A. Torralba, "Recognizing scene viewpoint using panoramic place representation", en *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, págs. 2695-2702. DOI: 10.1109/CVPR.2012.6247991.

[349] W. Yang, Y. Qian, J.-K. Kamarainen, F. Cricri y L. Fan, "Object Detection in Equirectangular Panorama", en *2018 24th International Conference on Pattern Recognition (ICPR)*, Beijing: IEEE, 2018, págs. 2190-2195, ISBN: 9781538637883. DOI: 10.1109/ICPR.2018.8546070. dirección: https://ieeexplore.ieee.org/document/8546070/.

[350] S. Anand y L. Priya, *A guide for machine vision in quality control*, 1.ª ed. Boca Raton: CRC Press, 2019, ISBN: 9780815349273.

[351] R. Prados, R. Garcia y L. Neumann, "State of the Art in Image Blending Techniques", en *Image Blending Techniques and their Application in Underwater Mosaicing*, ép. SpringerBriefs in Computer Science, Cham: Springer International Publishing, 2014, págs. 35-60, ISBN: 9783319055589. DOI: 10.1007/978-3-319-05558-9_3.

[352] D. Ghosh y N. Kaabouch, "A survey on image mosaicing techniques", *Journal of Visual Communication and Image Representation*, vol. 34, págs. 1-11, ene. de 2016, ISSN: 10473203. DOI: 10.1016/j.jvcir.2015.10.014.

[353] K. Simonyan y A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *arXiv:1409.1556 [cs]*, abr. de 2015, arXiv: 1409.1556. dirección: http://arxiv.org/abs/1409.1556.

[354] J. Cabrera, S. Cebollada, L. Payá, M. Flores y O. Reinoso, "A Robust CNN Training Approach to Address Hierarchical Localization with Omnidirectional Images:" en *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics*, Online Streaming: SCITEPRESS, 2021, págs. 301-310, ISBN: 9789897585227. DOI: 10.5220/0010574603010310. dirección: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010574603010310.